

ANALISA IMPLEMENTASI SENSOR GYROSCOPE PADA PERGERAKAN ROBOT HUMANOID BERBASIS OPEN CM 9.04

Refli Erdinal¹⁾, Feri Candra²⁾

¹⁾Mahasiswa Program Studi Teknik Elektro S1, ²⁾Dosen Teknik Elektro Program Studi Teknik Elektro S1, Fakultas Teknik Universitas Riau
Kampus Binawidya Jl. H.R Soebrantas Km 12,5 Simpang Baru Panam,
Pekanbaru, Riau 28293
E-mail : refli.erdinal7727@student.unri.ac.id

ABSTRACT

Humanoid robot itself is a robot that has human-like characteristics. Starting from the physical structure in general humanoid robots have a head, two hands, two legs, but there are some humanoid robots that are very humanoid such as having a waist to the upper body or even a face. . The purpose of this study is to allow the robot to move steadily and balancedly according to the willingness of the tester. This humanoid robot can perform walking, running, jumping, dancing, and much more. Simple gestures such as walking require alignment between actuators, controllers, and built-in programs or commands. It uses an OpenCM 9.04 controller and Gyroscope sensor as its balancing sensor. Gyroscope sensor used by MPU-6050 which has an output in the form of angular velocity from the direction of 3 axes that is, the x axis that will later become the angle of phi (right and left), and the y-axis will later become theta angle (top and bottom), and the z axis will later become psi angle (front and back). Gyroscope sensors will be implemented to the legs of humanoid robots so that humanoid robots will maintain their balance in standing and walking later. The robot used is a Bioloid Type-A humanoid robot. The results of this test the robot can move as expected by the author and add value to the test results of the programming process of the OpenCM 9.04 microcontroller.

Keywords: OpenCM 9.04, Gyroscope, MPU-6050, Bioloid Type-A humanoid robot.

I. PENDAHULUAN

Kontes Robot Indonesia (KRI) merupakan kompetisi pada bidang robot yang diadakan oleh Kementrian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia (RISTEKDIKTI) yang salah satu cabangnya yaitu Kontes Robot Seni Tari Indonesia (KRSTI) dengan tema seni tari.

Robot humanoid sendiri merupakan robot yang mempunyai karakteristik seperti manusia. Dimulai dari struktur fisik secara umum robot humanoid memiliki kepala, dua buah tangan, dua buah kaki, namun ada beberapa robot humanoid yang sangat menyerupai manusia seperti memiliki bagian pinggang hingga badan atas atau bahkan wajah (Cui dkk., 2013).

Tujuan utama menciptakan robot humanoid yaitu untuk meringankan pekerjaan manusia dimana robot dapat bekerjasama

sama dengan manusia berdasarkan fleksibilitas robot dan adaptasi robot terhadap alat atau benda dan lingkungan manusia (Monteiro dkk., 2012). Bagian penggerak pada robot humanoid terdiri dari susunan aktuator atau penggerak yang memiliki tingkat derajat kebebasan tertentu. Satu contoh aktuator yaitu aktuator robot dynamixel AX-12A dan dynamixel AX-18A. Untuk menggerakkan aktuator dibutuhkan kontroler. Salah satunya OpenCM 9.04. Terdapat tiga jenis OpenCM 9.04, yaitu OpenCM 9.04-A, OpenCM 9.04-B, dan OpenCM 9.04-C. Ketiganya memiliki fungsi yang sama yaitu memberikan perintah yang telah dibangun pada aktuator. Bahasa yang digunakan yaitu bahasa pemrograman C atau C++. Dalam penelitian ini OpenCM menggunakan pemrograman dengan aplikasi OpenCM IDE, dimana saat ini OpenCM IDE

tidak melakukan update lagi sehingga mempersulit pengujian untuk pemrograman.

Untuk menstabilkan robot humanoid dibutuhkan perangkat tambahan berupa sensor *Gyroscope*. Sensor *gyroscope* adalah perangkat untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut. Mekanismenya adalah sebuah roda berputar dengan piringan didalamnya yang tetap stabil. Sensor *gyroscope* yang digunakan MPU-6050 yang memiliki keluaran berupa kecepatan sudut dari arah 3 sumbu yaitu, sumbu x yang nantinya akan menjadi sudut *phi* (kanan dan kiri), dan sumbu y nantinya menjadi sudut *theta* (atas dan bawah), dan sumbu z nantinya menjadi sudut *psi* (depan dan belakang).

II. LANDASAN TEORI

2.1 Robot Humanoid

Robot humanoid yaitu robot yang penampilan semuanya dibentuk berlandaskan tubuh manusia, dan mampu melakukan interaksi dengan peralatan maupun sekeliling yang terkait yang dibuat untuk manusia. Secara umum robot humanoid memiliki tubuh dengan kepala, dua buah lengan dan dua kaki, meskipun hadir pula beberapa susunan robot humanoid yang hanya berupa beberapa dari tubuh manusia, misalnya dari pinggang ke atas. Beberapa robot humanoid juga memiliki wajah yang sempurna dengan mata dan mulut. Android yaitu robot humanoid susunan untuk secara estetika menyerupai manusia.

2.2 Komponen Dasar Robot Humanoid

- **Sensor**

Sensor yaitu alat yang dapat mengukur beberapa atribut dan yaitu salah satu dari tiga primitif dari robotika (disamping perencanaan dan pengendalian). Penginderaan memperagakan peran penting dalam paradigma robotika. Sensor yang digunakan dalam penelitian ini yaitu sensor *Gyroscope*. *Gyroscope* merupakan suatu alat elektronik yang berfungsi untuk mengukur kecepatan sudut dengan satuan ($^{\circ}/s$) yang dialami oleh suatu benda pada *pitch*, *roll* dan *yaw*. Sehingga dengan memanfaatkan data

kecepatan sudut tersebut dapat diketahui sudut kemiringan benda seperti Gambar 1. dibawah.



Gambar 1. Gyro+Accelerometer Sensor, 3 Axis based on MPU-6050

- **Aktuator (Penggerak)**

Aktuator berperan penting sebagai penggerak robot. Salah satu contoh aktuator yaitu servo dynamixel AX-12A. Servo tersebut memiliki kemampuan untuk mendeteksi kecepatan, suhu, posisi, tegangan, bahkan dapat memproteksi diri dengan mendeteksi kesalahan/error yang terjadi selama proses operasi berlangsung. Motor servo dynamixel yang digunakan pada Robot Bioloid menggunakan tipe AX-18A dan AX-12A. Perbedaan dari kedua tipe tersebut terletak pada torsi. Tipe AX-18A memiliki torsi yang lebih besar daripada tipe AX-12A seperti Gambar 2.



Gambar 2. Servo Dynamixel AX-12A dan Servo Dynamixel AX-18A (ROBOTIS)

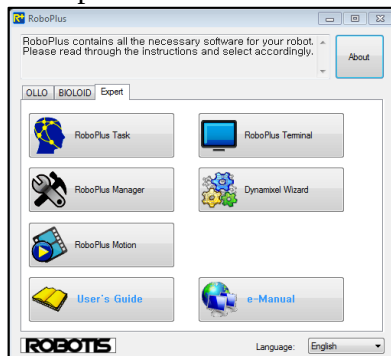
- **Program dan Kontroler**

Program dan kontroler merupakan faktor yang tidak bisa lepas satu sama lain. Program yang merupakan serangkaian algoritma yang ditanam pada kontroler. Untuk program, jika menggunakan kontroler bawaan dari robotis menggunakan CM 530 seperti Gambar 3.



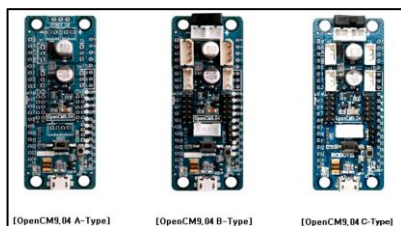
Gambar 3. CM-530 (ROBOTIS)

Untuk mengoperasikan CM 530, perlu menggunakan program yaitu RoboPlus. Roboplus merupakan *software* khusus yang memang diperuntukkan untuk produk buatan ROBOTIS seperti Gambar 4.



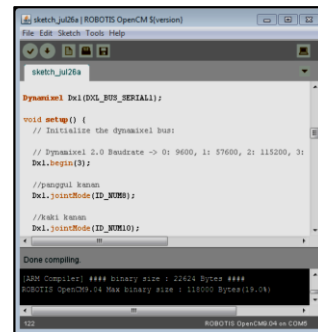
Gambar 4. RoboPlus

Kontroler akhir yang akan dipakai dalam penelitian yaitu OpenCM 9.04. Merupakan kontroler yang *compatible* dengan servo Dynamixel AX-12. OpenCM 9.04 memiliki 3 tipe yaitu OpenCM 9.04-A, OpenCM 9.04-B, dan OpenCM 9.04-C. Diantara ketiganya tipe C merupakan tipe yang paling lengkap secara fisik (konektor) seperti Gambar 5.



Gambar 5. OpenCM 9.04 Series (ROBOTIS)

Untuk OpenCM 9.04 dapat diprogram menggunakan program OpenCM IDE. Program ini bersifat *open-source* dilengkapi dengan berbagai macam *library* juga fitur *Example* sebagai contoh dasar untuk memprogram OpenCM 9.04 yang dapat dimodifikasi dengan mudah. Gambar 6 merupakan OpenCM IDE.



Gambar 6. OpenCM IDE

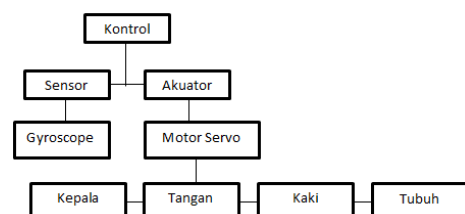
• Sumber Daya

Sumber daya yang digunakan yaitu menggunakan Baterai Lipo 3 sel (2S) seperti Gambar 7. Baterai lipo juga merupakan salah satu jenis baterai yang sering digunakan dalam dunia Robot Control, memiliki rating 3.7 volt per sel dengan total 3 sel Baterai yaitu 11.1 volt.



Gambar 7. Baterai Lipo

2.3 Kerangka Konsep Penelitian



Gambar 8. Diagram kerangka konsep penelitian

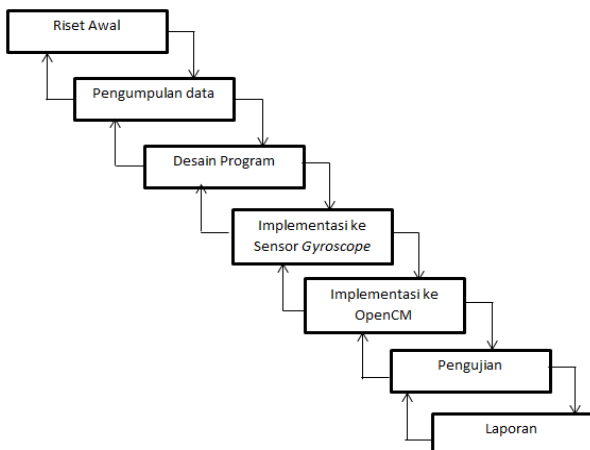
Pada gambar 8 diatas algoritma yang dibangun mengimplementasikan sensor *gyroscope* pada gerakan motor servo pada kepala, tangan, kaki, dan tubuh. Algoritma dibuat dengan pemrograman C++

menggunakan OpenCM IDE, dengan menggunakan kontroler OpenCM 9.04 sebagai akhir.

III. METODE PENELITIAN

3.1 Diagram Metode Penelitian

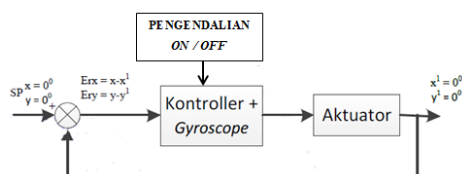
Metode penelitian ini menggunakan metode diagram *Waterfall* ditunjukkan pada Gambar 9.



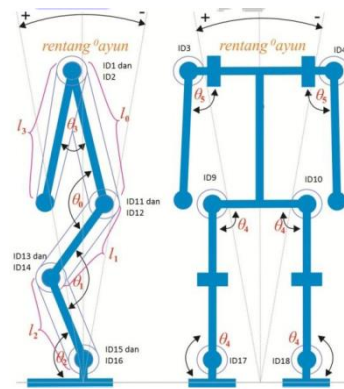
Gambar 9. Diagram Waterfall

3.2 Sistem Pengendalian Keseimbangan Robot

Robot diharuskan mempertahankan pose tubuh robot pada posisi tertentu dan mempertahankan kondisi kesetimbangan yang telah di tentukan. Dengan menjaga posisi robot tetap tegap, maka kondisi robot akan berada pada posisi yang lebih stabil, dan menjadi dasar untuk sistem kestabilan robot dalam mengkompensasi gangguan eksternal. Pegontrol *on/off* merupakan salah satu jenis aksi pengontrolan yang banyak digunakan pada pengontrol otomatis karena sederhana dan relatif murah. Berikut adalah diagram pengontrolan *on/off* dalam pengendalian keseimbangan robot pada Gambar 10 dan pose dinamik robot pada Gambar 11.



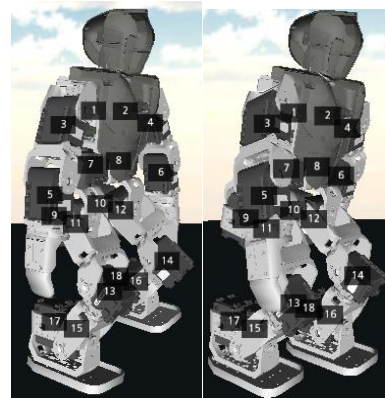
Gambar 10. Diagram pengendalian keseimbangan robot



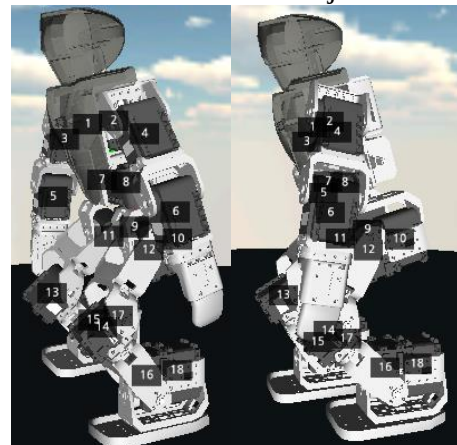
Gambar 11. Pose Dynamic Robot

3.3 Dinamika Gerak Robot

Gerakan robot berkaki dua memiliki kerumitan tersendiri dalam gerakan yang dinamis. Manusia yang sehat dapat menjaga keseimbangan saat berjalan karena keterampilan yang diperoleh dengan belajar. Di bawah keterampilan seperti itu, komponen kekuatan / momen yang terdapat pada fase kaki satu, atau pada kedua kaki seperti Gambar 12 dan 13.



Gambar 12. Dinamika gerak angkat kaki kiri dan kaki kiri maju



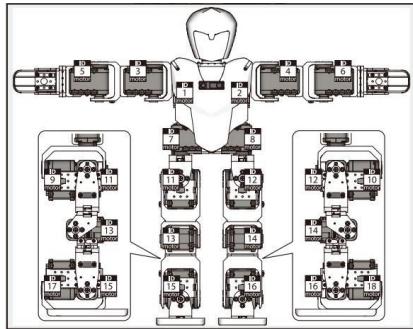
Gambar 13. Dinamika gerak angkat kaki kanan dan kaki kanan maju

IV. ANALISIS DAN PEMBAHASAN

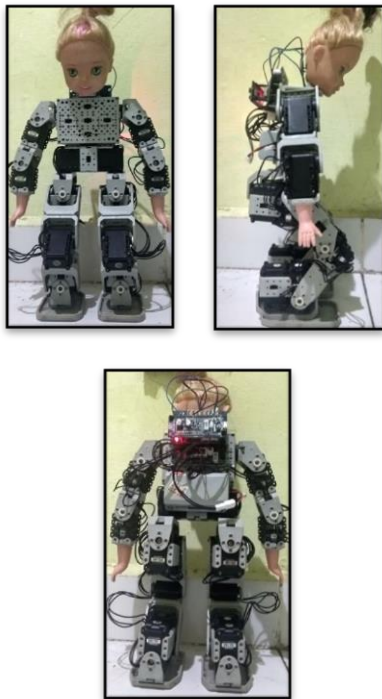
Dalam pembahasan ini akan menjelaskan tentang pengujian alat dan sistem yang telah dirancang dan dibuat, selain itu juga akan mencantumkan data hasil pengujian serta pembahasannya. Pengujian mekanik Robot Humanoid akan meliputi posisi servo pada robot, sensor *gyroscope*, dan kinerja Robot.

4.1 Struktur Robot

Bagian robot yang digunakan hanya bagian kaki hingga panggul menggunakan jenis robot Bioloid Tipe A. Jumlah *Degree of Freedom* (DoF) pada robot hingga panggul yaitu 18 servo. Struktur robot ditunjukkan pada Gambar 14,15, dan Tabel 1.



Gambar 14. Struktur Robot (ROBOTIS)



Gambar 15. Struktur Robot Bioloid tampak depan, samping, dan belakang

Tabel 1. Spesifikasi robot

<u>Tinggi</u>	40 cm
<u>Berat</u>	1.9 kg
<u>Jumlah sendi</u>	18
<u>Controller</u>	<u>OpenCM 9.04</u>
<u>Sensor</u>	MPU6050
<u>Sumber daya</u>	LIPO 11.1 V

4.2 Analisis Pada Pemograman OpenCM 9.04

Pengontrolan yang digunakan pada robot yaitu OpenCM 9.04, menggunakan bahasa C++. Pada saat pembuatan program terjadi beberapa masalah yang mempengaruhi penelitian, salah satunya kurang *updatenya* OpenCM IDE yang menyebabkan error pada pembuatan program dilihat dari Gambar 16, dan susahnya mengklarifikasi program dan kurangnya penelitian perancangan program OpenCM yang mempengaruhi berhasilnya penelitian ini.

The **ROBOTIS OpenCM** is a development Software and download tool for the OpenCM9.04 embedded board. Sources of the ROBOTIS OpenCM are released under licenses of their respective authors. Copyright (c) ROBOTIS Co., Ltd. Modified or newly-created codes are released under the GNU GPL or LGPL licenses. For more information on the OpenCM9.04 refer to the Appendix section of the e-manuals.

- GNU GPL
- GNU LGPL

WARNING : OpenCM IDE is not updated anymore. It is recommended to use **Arduino IDE** as OpenCM IDE isn't maintained anymore.

Gambar 16. OpenCM IDE tidak *update*

4.3 Analisis posisi 18 motor servo pada robot humanoid

Perancangan posisi servo pada robot humanoid di hubungkan secara berurutan dengan label yang sudah di indifikasi tiap servo. Posisi servo pada robot humanoid akan memberikan sebuah pergerakan pada robot sehingga robot akan bergerak sesuai perintah yang diberikan oleh mikrokontroler OpenCm9.04. Data nilai posisi setiap servo yang telah diberikan ke robot humanoid ditunjukkan pada Tabel 2.

Tabel 2. Data nilai servo robot dalam kondisi berdiri dan berjalan

No ID	Langkah Pergerakan Robot						
	0 s	0.08 s	0.16 s	0.24 s	0.32 s	0.40 s	0.48 s
	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC
1	235	235	235	235	235	235	235
2	788	788	788	788	788	788	788
3	279	279	279	279	279	279	279
4	744	744	744	744	744	744	744
5	462	462	462	462	462	462	462
6	561	561	561	561	561	561	561
7	358	358	358	358	358	358	358
8	666	666	666	666	666	666	666
9	507	507	507	507	507	507	507
10	516	516	516	516	516	516	516
11	341	341	341	341	341	341	341
12	682	682	682	682	682	682	682
13	240	240	240	240	240	240	240
14	783	783	783	783	783	783	783
15	647	647	647	647	647	647	647
16	376	376	376	376	376	376	376
17	507	507	507	507	507	507	507
18	516	516	516	516	516	516	516

No ID	Langkah Pergerakan Robot						
	0 s	0.08 s	0.16 s	0.24 s	0.32 s	0.40 s	0.48 s
	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC
1	233	248	263	276	287	295	301
2	786	801	816	829	840	848	854
3	279	279	279	279	279	279	279
4	744	744	744	744	744	744	744
5	462	462	462	462	462	462	462
6	561	561	561	561	561	561	561
7	358	358	358	358	358	358	358
8	666	666	666	666	666	666	666
9	503	521	528	524	519	513	507
10	555	544	537	533	528	522	516
11	353	357	355	352	353	357	364
12	725	718	713	718	720	719	716
13	267	264	259	252	248	246	248
14	807	765	744	751	760	769	775
15	641	648	652	655	660	666	671
16	386	421	437	434	427	418	409
17	533	531	528	524	519	513	507
18	544	541	537	533	528	522	516

No ID	Langkah Pergerakan Robot						
	0 s	0.08 s	0.16 s	0.24 s	0.32 s	0.40 s	0.48 s
	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC
1	300	303	302	298	290	280	268
2	853	856	855	851	843	833	821
3	279	279	279	279	279	279	279
4	744	744	744	744	744	744	744
5	462	462	462	462	462	462	462
6	561	561	561	561	561	561	561
7	358	358	358	358	358	358	358
8	666	666	666	666	666	666	666
9	501	496	490	482	473	467	465
10	510	505	504	510	519	527	530
11	363	372	373	361	341	319	299
12	721	715	709	702	696	691	688
13	254	262	262	245	222	203	196
14	777	775	771	766	761	757	756
15	675	675	677	682	685	681	669
16	402	398	395	394	393	392	390
17	501	496	491	486	482	480	479
18	510	505	500	496	493	491	490

No ID	Langkah Pergerakan Robot						
	0 s	0.08 s	0.16 s	0.24 s	0.32 s	0.40 s	0.48 s
	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC	Nilai ADC
1	254	239	223	209	209	184	175
2	807	792	776	762	762	737	728
3	279	279	279	279	279	279	279
4	744	744	744	744	744	744	744
5	462	462	462	462	462	462	462
6	561	561	561	561	561	561	561
7	358	358	358	358	358	358	358
8	666	666	666	666	666	666	666
9	467	473	482	490	490	501	507
10	527	519	510	504	504	510	516
11	287	283	286	291	291	293	297
12	686	685	684	683	683	676	669
13	203	222	245	262	262	254	248
14	757	761	766	771	771	777	775
15	649	627	607	595	595	605	614
16	387	382	376	369	369	357	352
17	480	482	486	491	491	501	507
18	491	493	496	500	500	510	516

4.4 Pengujian sensor MPU 6050

Pada pengujian ini digunakan sensor MPU6050 untuk mengumpulkan data sudut masing masing gerakan pada robot, agar hasil pergerakan robot menjadi stabil dan seimbang. Data nilai pergerakan robot sebelum diimplementasikan oleh *gyroscope* ditunjukkan oleh Tabel 3, Tabel 4, Tabel 5, Tabel 6, Tabel 7, dan Tabel 8.

Tabel 3. Data nilai servo robot dalam kondisi berdiri dan berjalan

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	0	0	0	5	4.99	5	5
Pitch	3	3	3	4.46	4.47	4.48	1.48
Yaw	0	0	0	0.18	0.19	0.19	0.19

Tabel 4. Data nilai *gyroscope* robot dalam kondisi angkat kaki kiri

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	-8	-8	-8	9.60	9.61	9.62	17.62
Pitch	26	26	26	3.06	3.07	3.08	22.92
Yaw	-1	-1	-1	-1.75	-1.76	-1.77	-0.77

Tabel 5. Data nilai *gyroscope* robot dalam kondisi kaki kiri maju

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	-4	-4	-4	10.18	10.19	10.20	14.20
Pitch	3	3	3	4.50	4.51	4.52	1.52
Yaw	19	19	19	19.48	19.49	19.50	0.50

Tabel 6. Data nilai *gyroscope* robot dalam kondisi angkat kaki kanan

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	8	8	8	7.69	7.69	7.70	0.30
Pitch	24	24	24	1.75	1.76	1.76	22.24
Yaw	2	2	2	2.41	2.41	2.42	0.42

Tabel 7. Data nilai *gyroscope* robot dalam kondisi kaki kanan maju

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	6	6	6	13.96	13.96	13.98	7.98
Pitch	3	3	3	1.90	1.90	1.92	1.08
Yaw	-19	-19	-19	-19.74	-19.74	-19.75	-0.75

Tabel 8. Data kesalahan nilai *gyroscope* pada robot dalam berjalan

Sudut (°)	Langkah Pergerakan Robot (Real)				Offset (°)
	Angkat kaki kiri	Kaki kiri maju	Angkat kaki kanan	Kaki kanan maju	
Roll	17.62	14.20	0.30	7.98	18 sampai 0
Pitch	22.92	1.52	22.24	1.08	23 sampai 1
Yaw	-0.77	0.50	0.42	-0.75	0.5 sampai -1

4.5 Implementasi robot

Robot yang digunakan adalah Bioloid tipe-A yang terdiri dari 18 buah servo AX-12A. Pada saat robot berdiri diam hasil data yang diberikan sensor MPU6050 mengalami kesalahan *x/roll* sebesar 5° , *y/pitch* sebesar 1.48° , dan *z/yaw* sebesar 0.19° , tetapi robot tetap stabil saat berdiri. Pada saat robot mulai berjalan nilai *x/roll* melakukan kesalahan *offset* antara 18 sampai 0 derajat, nilai *y/pitch* melakukan kesalahan *offset* antara 23 sampai 1 derajat, dan nilai *z/yaw* melakukan kesalahan *offset* antara 0.5 sampai -1 derajat sehingga menyebabkan robot berjalan miring ke kanan dan kiri. Agar robot dapat berjalan dengan stabil dengan sedikit kesalahan sensor MPU akan di implementasikan dalam proses pergerakan robot. Hasil dari analisis robot yang terjadi saat melakukan pergerakan dengan menggunakan sensor *gyroscope* dimuat dalam Tabel 9, Tabel 10, Tabel 11, Tabel 12, Tabel 13, Tabel 14, dan Tabel 15.

Tabel 9. Data nilai *gyroscope* robot dalam kondisi berdiri

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	0	0	0	3.30	3.27	3.32	-3.32
Pitch	3	3	3	3.55	3.55	3.55	-0.55
Yaw	0	0	0	-0.40	-0.38	-0.40	0.4

Tabel 10. Data nilai *gyroscope* robot dalam kondisi angkat kaki kiri

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	-8	-8	-8	-8.13	-8.12	-8.14	0.14
Pitch	26	26	26	-3.08	-3.08	-3.08	-29.02
Yaw	-1	-1	-1	-1.35	-1.34	-1.35	0.35

Tabel 11. Data nilai *gyroscope* robot dalam kondisi kaki kiri maju

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	-4	-4	-4	-3.66	-3.65	-3.66	-0.44
Pitch	3	3	3	3.03	3.04	3.03	-0.03
Yaw	19	19	19	18.34	18.50	18.34	0.66

Tabel 12. Data nilai *gyroscope* robot dalam kondisi angkat kaki kanan

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	8	8	8	8.04	8.10	8.11	-0.11
Pitch	24	24	24	12.63	12.65	12.70	11.30
Yaw	2	2	2	2.31	2.28	2.28	-0.28

Tabel 13. Data nilai *gyroscope* robot dalam kondisi kaki kanan maju

Sudut (°)	Langkah Pergerakan Robot (Invers Kinematic)			Langkah Pergerakan Robot (Real)			Error
	0 s	1 s	2 s	0 s	1 s	2 s	
Roll	6	6	6	6.73	6.71	6.73	-0.73
Pitch	3	3	3	2.60	2.61	2.62	0.38
Yaw	-19	-19	-19	-20.48	-20.47	-20.48	1.48

Tabel 14. Data kesalahan nilai *gyroscope* pada robot dalam berjalan

Sudut (°)	Langkah Pergerakan Robot (Real) (°)				Offset (°)
	Angkat kaki kiri	Kaki kiri maju	Angkat kaki kanan	Kaki kanan maju	
Roll	0.14	-0.44	-0.11	-0.73	1 sampai -1
Pitch	-29.02	-0.03	11.30	0.38	11 sampai -30
Yaw	0.35	0.66	-0.28	1.48	2 sampai -1

Tabel 15. Data perbandingan *offset* dengan *gyroscope* dan tidak ada *gyroscope*

Sudut (°)	Langkah Pergerakan <i>Offset</i> Robot (Real) (°)		Reduction <i>Offset</i> (°)
	Tidak ada <i>gyroscope</i>	Dengan <i>gyroscope</i>	
Roll	18 sampai 0	1 sampai -1	17
Pitch	23 sampai 1	11 sampai -30	12
Yaw	0.5 sampai -1	2 sampai -1	-1.5

Dilihat dari hasil pengujian saat sensor MPU6050 di implentasikan keadaan robot menjadi sedikit lebih stabil dibandingkan saat belum memakai sensor.

4.6 Hasil Pergerakan Robot

Dari proses penelitian yang dilakukan, dapat diambil data hasil dari pengujian pergerakan robot dalam Tabel 16 dibawah.

Tabel 16. Data hasil pergerakan robot

Gerakan	Kondisi Robot			
	0s	2s	4s	6s
Berdiri diam	Baik	Baik	Baik	Baik
Berjalan	Baik	Baik	Kurang	Kurang
Menari	Baik	Baik	Baik	Baik

Pada saat pengujian, robot dapat bertahan selama kurang lebih 3 menit, setelah itu servo bagian kaki akan panas dan mengalami beberapa kesalahan saat berjalan.

4.7 Faktor yang mempengaruhi keseimbangan robot

Saat proses pergerakan robot, banyak sekali faktor yang mempengaruhi keseimbangan robot yaitu :

- Sumber daya pada *controller* yang tidak stabil akan menyebabkan robot tidak dalam kondisi yang bagus sehingga menyebabkan pergerakan robot terhalang dan tidak seimbang. Disini penguji memakai baterai Lipo dengan tegangan 11.1 Volt, tegangan yang dibutuhkan robot untuk bergerak membutuhkan 12 Volt keatas sehingga keadaan robot tidak stabil.
- *Controller* yang digunakan penguji yaitu OpenCM 9.04 sering mengalami error dan dapat menyebabkan robot tidak bergerak sesuai perintah.

- Servo yang digunakan dalam pengujian yaitu servo AX-12A dan AX-18A tidak mampu menahan beban robot lebih lama sehingga servo akan panas dan menyebabkan robot kehilangan kendali.

- Keadaan statis dari tempat pengujian juga menjadi faktor robot tidak seimbang, saat robot berjalan dalam keadaan licin dan tempat tidak rata akan menyebabkan kehilangan kendali.

V. KESIMPULAN

Berdasarkan hasil dan pengujian dan analisa dari pergerakan robot bioloid berjalan, dalam pergerakan robot, sensor *gyroscope* MPU6050 dapat digunakan sebagai pendeteksi kemiringan dari robot humanoid. Data yang didapat dari sensor MPU6050 adalah *roll*, *pitch*, dan *yaw* atau X, Y , dan Z yang merupakan data dari kemiringan robot. Data kemiringan robot yang mengalami perubahan selama robot berjalan adalah *roll* dan *pitch* atau X dan Y sebesar 18 sampai 0 derajat dan 23 sampai 1 derajat. Setelah robot di implementasikan oleh sensor MPU6050 data kemiringan *roll* dan *pitch* menjadi sedikit lebih stabil sebesar 1 sampai -1 derajat dan 11 sampai -30 derajat. Sehingga robot berjalan sedikit keluar dari jalur yang telah ditetapkan.

Penggunaan dari *Controller* OpenCM 9.04 menjadi faktor penting juga dalam pengembangan robot karena masih belum banyak penelitian yang dilakukan dengan *controller* tersebut. Masih minimnya pengerjaan program yang bisa dilakukan oleh OpenCM 9.04 menjadi hambatan yang mengakibatkan kerugian dalam pengujian dari penelitian ini.

Penggunaan dari servo AX-12A dan AX-18A yang tidak dapat menahan beban robot lebih lama juga dapat menjadi faktor yang mempengaruhi keseimbangan robot.

DAFTAR PUSTAKA

- Abdolshah, S., Abdolshah, M., & S.H.Tang. (2015). Trajectory Planning and Walking Pattern Generation of Humanoid Robot Motion. *International Journal of Robotics and Automation (IJRA)*, 4(2), 135~142.
- Cui, Y., Shon, S. G., & Byun, H. J. (2013). Control Structure Design for Man-Function Humanoid. *TELKOMNIKA*, 11(12), 7745~7753.
- Indonesia, Kontes Robot Indonesia (KRI). (2019). Panduan Kontes Robot Seni Tari Indonesia (KRSTI) 2019, from <http://kontesrobotindonesia.id/data/2019/PanduanUmum KRI2019>.
- Lukas B.S, Deddy S, dan Dede I. (2014). “Sistem Kendali Gerak Segway Berbasis Mikrokontroler”. *Techné Jurnal Ilmiah Elektroteknika*, 13(1), 125~134. Universitas Kristen Satya Wacana, Salatiga.
- Monteiro, R., Reis, L. P., & Pereira, A. C. (2012). Humanoid Dynamic Controller. *TELKOMNIKA*, 10(8), 1936~1947.
- Muhammad K. F, Barlian H. P, dan Rizal M. (2017). “Perancangan Dan Implementasi *Real Segway* Pada *Skateboard* Roda Satu Menggunakan *Gyroscope* Dan *Accelerometer*” dalam Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Volume 1 No 1 (hlm. 48-56). Universitas Brawijaya.
- Raranda, dan Puput W.R. (2017). “Implementasi Kontroler PID Pada *Two Wheels Self Balancing* Robot Berbasis Arduino UNO”. *Jurnal Jurusan Teknik Elektro*. 6(2), 89 ~96. Universitas Negeri Surabaya.
- ROBOTIS. ROBOTIS e-Manual, 2018, from <http://support.robotis.com/en/>