

PERANCANGAN ROBOT BERKAKI 4 (*QUADRUPED*) DENGAN *STABILIZATION ALGORITHM* PADA *UNEVEN FLOOR* MENGUNAKAN 6-DOF IMU BERBASIS *INVERS KINEMATIC*

Rofiq Cahyo Prayogo^{*)}, Aris Triwiyatno, dan Sumardi

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail : rofiqcp@gmail.com

Abstrak

Teknologi robotika selalu mengalami perkembangan dengan sangat cepat. salah satunya teknologi robot berkaki 4 (Quadruped). Salah satu jenis robot itu adalah robot berkaki 4 (Quadruped). Permasalahan yang sering timbul pada robot berkaki 4 ini adalah ketika dihadapkan pada permukaan yang tidak rata (uneven floor). Ini akan mengakibatkan pergerakan robot terhambat dikarenakan titik beban robot yang tidak seimbang. Pada penelitian ini dilakukan perancangan dan pembuatan algoritma stabilisasi (Stabilization Algorithm) robot quadruped pada uneven floor dengan sensor 6 DOF IMU MPU 6050 berbasis invers kinematic. Sensor MPU 6050 terbagi atas sensor accelerometer dan gyroscope. Sensor ini akan memberikan masukan sudut kemiringan robot sehingga akan diolah dalam stabilization algorithm yang menggunakan kontrol proportional untuk dapat mempertahankan body robot tetap datar pada uneven floor. Robot quadruped bergerak dengan menggunakan metode inverse kinematics dengan mengikuti koordinat yang diberikan. Metode inverse kinematics membantu menemukan nilai sudut yang menggerakkan servo melalui koordinat akhir. Hasil pengujian menunjukkan bahwa robot quadruped dengan menggunakan stabilization algorithm dapat meningkatkan kestabilan sudut pitch sebesar 75,58 % dan kestabilan sudut roll sebesar 71,15 %.

Kata kunci : Quadruped, Stabilization Algorithm, Invers Kinematic

Abstract

Robotics technology is always growing very quickly. One of type of robot is 4 legged robot (Quadruped). The Problem from this robot is if quadruped find the uneven floor. This is will result the movement of robot can be late and the center of gravity not balanced. In this research makes a designed stabilization algorithm quadruped on uneven floor with 6 DOF IMU MPU 6050 based on invers kinematic. MPU 6050 sensor devided by accelerometer and gyroscope. This sensor will gave the input of tilt degree of robot so can be processed in stabilization algorithm with proportional controller to can balance the body of robot on uneven floor. Quadruped robot moves using inverse kinematics by following the coordinates given. The method of inverse kinematics help find the value angle servo moves through the final coordinates. The test result shows that quadruped robot by using stabilization algorithm can increase stability of pitch angle equal to 75,58% and stability of roll angle equal to 71,15%.

Keyword : Quadruped, Stabilization Algorithm, Invers Kinematic

1. Pendahuluan

Robot merupakan barang yang sudah tidak asing lagi bagi mahasiswa elektro khususnya konsentrasi Kontrol dan Instrumentasi. Sejalan dengan perkembangan IPTEK yang pesat di semua bidang dan perubahan masyarakat yang dinamis terutama pada negara negara maju maupun yang sedang berkembang, penggunaan robot sangat diperlukan. Perkembangan robotika saat ini sangat pesat dengan berbagai macam system baru yang ditemukan. Salah satu jenis robot yang juga banyak dikembangkan adalah robot berkaki 4 (*Quadruped*). Robot ini bergerak berdasarkan kinematika gerak yang diterapkan pada tiap-tiap kaki yang disusun oleh motor-motor *servo* ^[1].

Beberapa kekurangan dari robot *quadruped* adalah ketika menemui permukaan lantai yang tidak rata (*uneven floor*), maka robot akan cenderung berjalan tidak stabil. Selain itu ketidakstabilan ini akan menyebabkan titik beban *body* robot *quadruped* akan berpindah dan mengakibatkan pembebanan pada salah satu motor servo. Hal ini akan menyebabkan motor servo yang diberikan beban tertinggi mengalami kerusakan yang lebih cepat ^[2].

Oleh karena itu pengontrolan *body* robot yang dapat mempertahankan posisi tetap datar diperlukan sehingga dalam kondisi lantai tidak rata (*uneven floor*), robot *quadruped* dapat menstabilisasikan *body* robot. Salah satu

cara untuk dapat membuat titik beban tetap stabil adalah dengan mengembangkan *invers kinematic*.

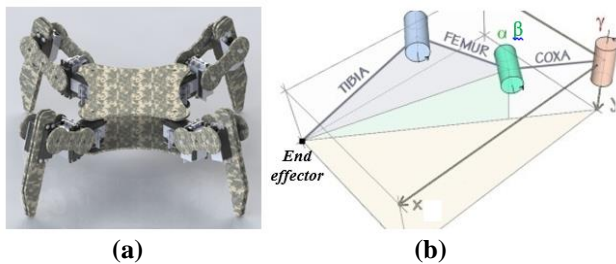
Invers Kinematic merupakan ragam gerak robot berorientasi jarak tempuh lengan. Kinematika ini akan menghasilkan pergerakan berbasis sudut translasi X, Y, Z dan sudut rotasi *Pitch, Roll, Yaw*. Sudut-sudut inilah yang akan dikontrol [3].

Selain kinematika gerak, diperlukan sensor yang dapat menghasilkan data sudut sebagai pembanding dan memberikan data *setpoint* yang kita inginkan. Salah satu sensor itu adalah IMU 6 DOF MPU 6050 [4]. Sensor ini merupakan gabungan 2 sensor, yaitu *Accelerometer* dan *Gyroscope*. Metode kontrol *Proportional* digunakan untuk mempertahankan robot pada *setpoint* yang diinginkan [5]. Kemudian hasil nilai kontrol *proportional* digunakan pada algoritma stabilisasi (*Stabilization Algorithm*) untuk mengubah ketinggian kaki robot. Dengan demikian diharapkan robot dapat otomatis menyesuaikan kakinya untuk dapat melewati lantai yang tidak rata (*uneven floor*).

2. Metode

2.1. Perancangan Perangkat Keras

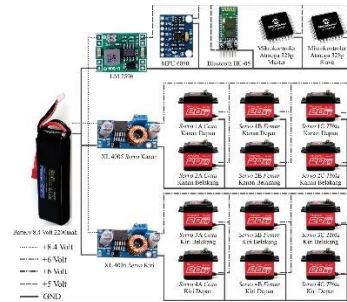
Perancangan spesifikasi mekanik robot *quadruped* ini dirancang dengan menggunakan 12 DoF (*Degree of Freedom*) dengan dibagi menjadi dua kaki setiap kaki terdapat 3 DoF sebagai penggerak menuju koordinat. Robot *quadruped* ini memiliki spesifikasi dengan panjang *link coxa* (L0) 53,7 mm, *link femur* (L1) 54,6 mm, dan *link tibia* (L2) 130 mm. Dari spesifikasi diatas, robot dapat diilustrasikan seperti pada Gambar 1.



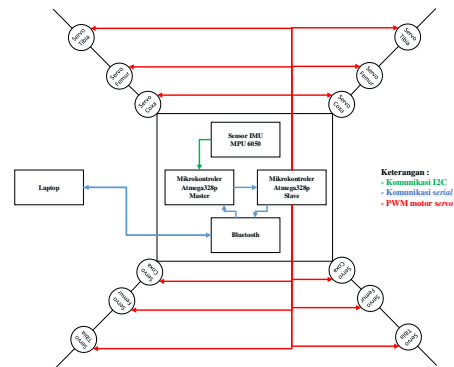
Gambar 1. Contoh robot *quadruped*, (a) Robot *quadruped*, (b) Konfigurasi *coxa, femur, dan tibia*.

Perangkat elektronik robot *quadruped* ini dirancang agar dapat bebas digunakan dimana saja, sehingga digunakan catu daya dari baterai. *Hardware* yang dirancang pada robot *quadruped* ini membutuhkan suplai tegangan yang berbeda, sehingga diperlukan regulator penurun tegangan sebanyak 2 macam dan disesuaikan dengan kebutuhan. Untuk catu daya mikrokontroler dan bluetooth digunakan penurun tegangan LM2596 karena hanya membutuhkan arus kurang dari 3A dan hanya membutuhkan tegangan kisaran 5V. Sedangkan untuk catu daya motor *servo* digunakan penurun tegangan XL4005 karena pada setiap

motor *servo* dan tegangan 6V agar dapat bekerja secara maksimal. Pembagian catu daya dapat dilihat pada Gambar 2. Perancangan input output perangkat keras robot *quadruped* ditunjukkan oleh blok diagram pada Gambar 3. Robot *quadruped* memiliki sensor IMU 6050 untuk pembacaan kemiringan *body*, *bluetooth* untuk komunikasi serial dengan laptop, dan 12 motor *servo* sebagai actuator robot.



Gambar 2. Catu daya robot *quadruped*



Gambar 3. Blok diagram input output robot *quadruped*

2.2. Sensor IMU MPU6050

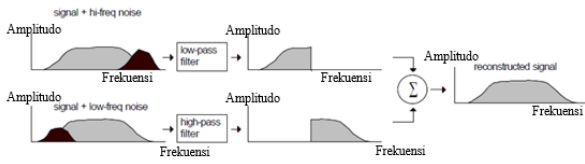
Sensor MPU-6050 merupakan gabungan dari 2 macam sensor yaitu sensor *accelerometer* dan *gyroscope* yang diproduksi oleh invensense dengan antarmuka komunikasi I2C[3]. MPU-6050 ini menggabungkan antara *accelerometer* dan *gyroscope* di dalam satu board. Sensor ini dapat mendeteksi percepatan dalam 3 axis (x, y, dan z) dan kecepatan sudut dalam 3 axis (x, y, dan z).



Gambar 4. Bentuk fisik sensor MPU6050.

Untuk mendapatkan data IMU MPU 6050 secara bagus maka diperlukan sebuah tapis atau *filter*. Dalam perancangan ini digunakan tapis komplementer yang direpresentasikan pada Gambar 5, menunjukkan bahwa

terdapat dua buah masukan yang salah satunya memiliki derau dengan frekuensi tinggi dan masukan yang lain memiliki frekuensi derau rendah. Masukan yang memiliki frekuensi derau tinggi ditapis dengan tapis pelewat-rendah, sedangkan masukan yang memiliki frekuensi derau rendah ditapis dengan tapis pelewat-tinggi. Hasil rekonstruksi kedua sinyal yang masing-masing telah ditapis tersebut merupakan variabel yang ingin dibaca tanpa adanya derau yang sebelumnya terasosiasi dengan sensor^[4].

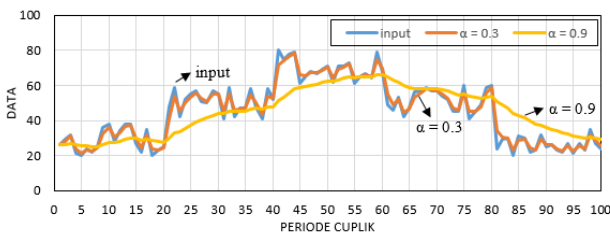


Gambar 5. Prinsip kerja tapis komplementer.

Exponential smoothing merupakan metode yang dapat digunakan untuk membuat suatu deretan data yang berurutan waktu menjadi lebih halus. Data yang diperhalus biasanya data yang mempunyai derau atau data acak. Data berurutan waktu biasa dilambangkan dengan x_t sedangkan data yang telah diperhalus dilambangkan dengan s_t dengan t adalah urutan waktu. Perhitungan Exponential smoothing dapat dituliskan seperti persamaan (2).

$$s_0 = x_0 \tag{1}$$

$$s_t = (1 - \alpha)x_t + \alpha s_{t-1}, t > 0 \tag{2}$$



Gambar 6. Pengaruh tapis exponential smoothing.

2.3. Kinematika Gerak

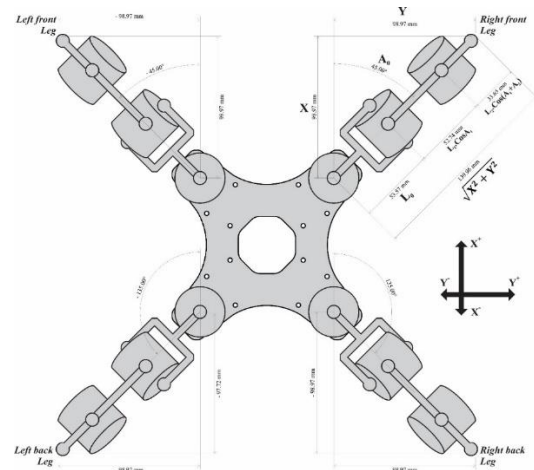
Untuk menunjang algoritma yang sudah dirancang maka diperlukan kinematika gerak untuk menggerakkan robot quadruped. Penyusun kinematika gerak dari quadruped merupakan inverse kinematic, motion planning, dan gait.

2.3.1. Invers Kinematic

Untuk mempermudah pengguna dalam menggerakkan end-effector dari robot lengan ke posisi tertentu, maka diperlukan metode gerak inverse kinematic. Metode ini menggunakan masukan posisi koordinat berupa X, Y, dan Z. Hasil keluaran dari inverse kinematic merupakan sudut yang terbentuk pada joint. Pada Gambar 7 dan Gambar 8 bahwa robot quadruped memiliki 3 joint pada tiap kakinya. Sudut yang dibentuk oleh setiap joint tersebut, yaitu sudut coxa (A0), sudut femur (A1), sudut tibia (A2). Lalu nilai

sudut-sudut tersebut dikonversi menjadi nilai PWM untuk menggerakkan servo.

Fungsi invers kinematic yang dirancang membutuhkan input koordinat X, Y, dan Z. Besaran X merupakan arah sumbu koordinat yang memiliki arah ke depan dan ke belakang robot. Besaran Y merupakan arah sumbu koordinat yang memiliki arah ke kanan dan ke kiri robot. Besaran Z merupakan arah sumbu koordinat yang memiliki arah ke atas dan ke bawah robot.

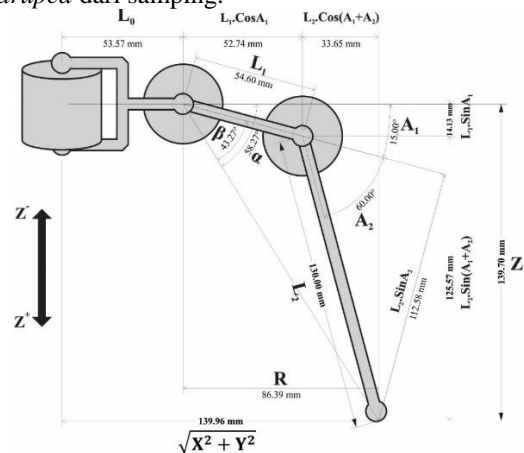


Gambar 7. Robot quadruped (top view).

Pembahasan rumus sudut yang dihasilkan dari fungsi invers kinematic akan dijelaskan pada bagian kaki kanan depan (right front leg) robot quadruped. Pada awalnya akan dimasukan nilai koordinat X, Y, dan Z. Untuk mendapatkan ketiga sudut pada joint maka digunakan pendekatan aljabar dan geometri. Perhitungan yang pertama dilakukan adalah mencari nilai sudut coxa (A0), yaitu menggunakan arcus tangen pada persamaan (3).

$$A_0 = \tan^{-1}\left(\frac{Y}{X}\right) \tag{3}$$

Untuk mencari sudut femur (A1) dan sudut tibia (A2), maka dijelaskan dengan Gambar 8 yang memperlihatkan robot quadruped dari samping.



Gambar 8. Robot quadruped (side view).

Selanjutnya dicarilah nilai resultan besaran dari X dan Y dengan persamaan (4)

$$\overline{XY} = \sqrt{X^2+Y^2} \quad (4)$$

Dikarenakan pengaruh dari mekanik yang tidak dapat membuat sudut *coxa* (A_0) dan sudut *femur* (A_1) menjadi satu titik, maka pada perhitungan menyebut *offset* atau pada Gambar 8 disebut L_0 . L_0 merupakan jarak dari *joint coxa* ke *joint femur* secara tegak lurus. Pada perhitungan selanjutnya disebut sebagai R, yaitu \overline{XY} dikurangi dengan L_0 , seperti pada persamaan (5)

$$R = \sqrt{X^2 + Y^2} - L_0 \quad (5)$$

Diketahui bahwa terdapat persamaan dalam *Forward Kinematic* sesuai dengan Gambar 8 diatas diperoleh nilai untuk titik R dan Z sebagai berikut :

$$R = L_1 \cos(A_1) + L_2 \cos(A_1 + A_2) \quad (6)$$

$$Z = L_1 \sin(A_1) + L_2 \sin(A_1 + A_2) \quad (7)$$

Untuk mendapatkan keluaran berdasarkan jarak yang diinginkan, maka perhitungan akan mencari nilai A_1 dan A_2 dengan mengkuadratkan R dan Z pada persamaan (6) dan (7) didapat :

$$[R]^2 = [L_1 \cos(A_1) + L_2 \cos(A_1 + A_2)]^2 \quad (8)$$

$$R^2 = L_1^2 \cos^2(A_1) + 2L_1L_2 \cos(A_1) \cos(A_1 + A_2) + L_2^2 \cos^2(A_1 + A_2) \quad (9)$$

$$[Z]^2 = [L_1 \sin(A_1) + L_2 \sin(A_1 + A_2)]^2 \quad (10)$$

$$Z^2 = L_1^2 \sin^2(A_1) + 2L_1L_2 \sin(A_1) \sin(A_1 + A_2) + L_2^2 \sin^2(A_1 + A_2) \quad (11)$$

Menjumlahkan R dan Z yang sudah di kuadratkan di persamaan (9) dan (11):

$$R^2 + Z^2 = L_1^2 [\sin^2(A_1) + \cos^2(A_1)] + L_2^2 [\sin^2(A_1 + A_2) + \cos^2(A_1 + A_2)] + 2L_1L_2 [\sin(A_1) \sin(A_1 + A_2) + \cos(A_1) \cos(A_1 + A_2)] \quad (12)$$

Dimana seperti yang kita ketahui bahwa :

$$\sin^2(\alpha) + \cos^2(\alpha) = 1 \quad (13)$$

Didapat persamaan yang lebih sederhana dari persamaan (12) yaitu :

$$R^2 + Z^2 = L_1^2 + L_2^2 + 2L_1L_2 [\sin(A_1) \sin(A_1 + A_2) + \cos(A_1) \cos(A_1 + A_2)] \quad (14)$$

Diperoleh persamaan sebagai penyederhanaan ke-2 rumus invers kinematic sebagai berikut :

$$\frac{R^2 + Z^2 - L_1^2 - L_2^2}{2L_1L_2} = \sin(A_1) \sin(A_1 + A_2) + \cos(A_1) \cos(A_1 + A_2) \quad (15)$$

Perhitungan selanjutnya didapat :

$$\cos(A_2) = \frac{R^2 + Z^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (16)$$

Dari persamaan (16) menghasilkan persamaan akhir untuk sudut *tibia* (A_2) sebagai berikut :

$$A_2 = \cos^{-1} \left\{ \frac{R^2 + Z^2 - L_1^2 - L_2^2}{2L_1L_2} \right\} \quad (17)$$

Kemudian lihat Gambar 8 kembali, dengan memperhatikan sudut *alpha* (α) dan sudut *gamma* (β), menghasilkan persamaan sebagai berikut :

$$\alpha = A_2 + \beta \quad (18)$$

dimana sudut *alpha* (α) dapat dihitung dari persamaan berikut :

$$\alpha = \tan^{-1} \frac{Z}{R} \quad (19)$$

dan sudut *gamma* (β) dapat dihitung dari persamaan berikut :

$$\beta = \tan^{-1} \frac{(L_2 \sin A_2)}{(L_1) + (L_2 \cos A_2)} \quad (20)$$

Dari persamaan (18), (19), dan (20) menghasilkan persamaan akhir untuk sudut *femur* (A_1) sebagai berikut :

$$A_1 = (\tan^{-1} \frac{Z}{R}) - (\tan^{-1} \frac{(L_2 \sin A_2)}{(L_1) + (L_2 \cos A_2)}) \quad (21)$$

2.3.2. Motion Planning

Motion planning merupakan sebuah proses untuk memecah nilai koordinat akhir yang ditentukan agar pergerakan servo lebih halus. Pergerakan servo LF-20MG untuk menjangkau sudut 60° adalah 0.16 detik. Motion planning ini berfungsi untuk memperlambat waktu pergerakan servo tersebut.

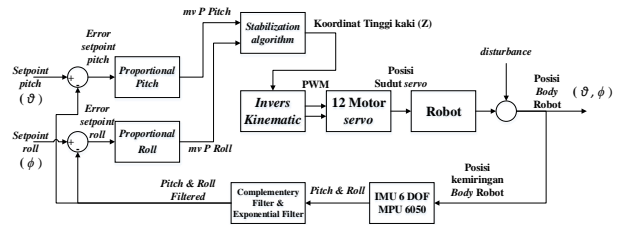
2.3.3. Perancangan Gait

Agar robot *quadruped* dapat berjalan maka diperlukan koordinat untuk menggerakkan end-effector berupa telapak kaki, susunan beberapa koordinat yang disebut gait. Fase pergerakan *quadruped* bahwa fase pergerakan *quadruped* dibagi menjadi 4 bagian:

1. Double Support (DS)
2. Pre-Swing
3. Single Support
4. Post-Swing

2.4. Stabilization Algorithm Robot Quadruped

Robot *quadruped* ini menggunakan *stabilization algorithm*, yang berisi algoritma dan kontroler PID. Nilai parameter-parameter PID diperoleh hasil *trial and error* dari data sensor yang sebelumnya sudah diuji pada beberapa percobaan. Secara umum, blok diagram proses kontrol robot *quadruped* ditunjukkan oleh Gambar 9.

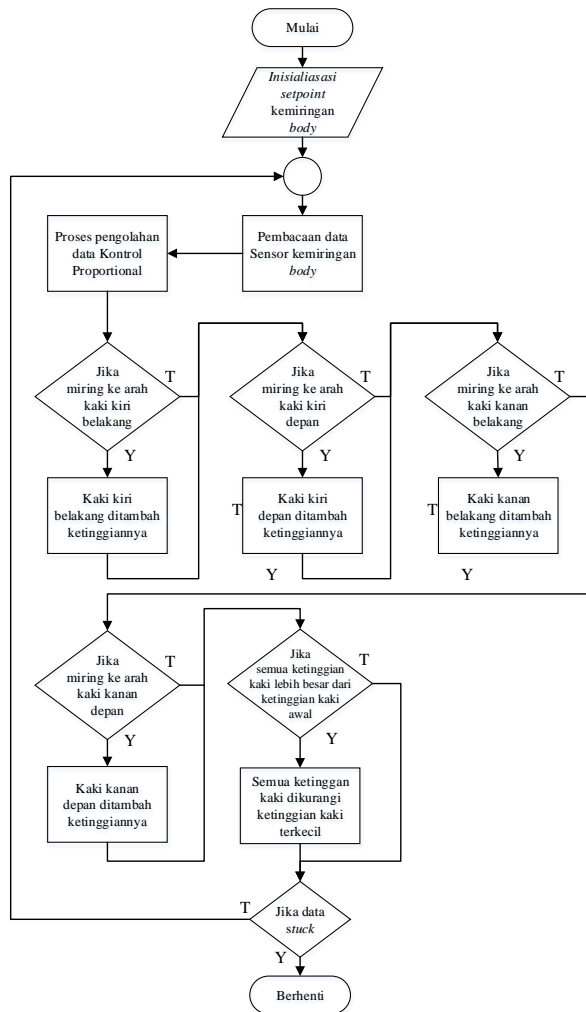


Gambar 9. Blok Diagram kontroler robot *quadruped*

Dalam menyetabilkan *body robot quadruped* maka dirancanglah sebuah algoritma stabilisasi (*Stabilization Algorithm*). *Stabilization Algorithm* ini sebelum perancangan program ke robot maka diperlukan sebuah flowchart untuk mempermudah memprogram robot

quadruped. Flowchart *Stabilization Algorithm* dapat dilihat pada Gambar 10.

Pada Gambar 10. dapat dilihat bahwa penjelasan secara simpelnya robot *quadruped* akan melakukan penyetabilan *body* dengan cara meninggikan masing-masing kaki robot satu persatu. Hal tersebut telah dirancang supaya robot *quadruped* dapat menyesuaikan pada lintasan *uneven floor*.



Gambar 10. Gambar Flowchart Stabilization Algorithm Robot Quadruped

3. Hasil dan Analisis

3.1. Pengujian Motor Servo

Pengujian motor servo dilakukan untuk mengetahui respon motor servo terhadap masukan sudut yang diberikan. Menghasilkan nilai error 0,17°.

3.2. Pengujian Invers Kinematic

Pengujian *inverse kinematics* dilakukan untuk menguji kepresisian dari robot *quadruped* dengan menggerakkan salah satu kakinya. Pada pengujian *inverse kinematics* ini

berisi pengujian variasi sumbu X, variasi sumbu Y, dan variasi sumbu Z.

Pengujian *inverse kinematics* variasi sumbu X berisi tentang pengujian metode *inverse kinematics* dengan variasi perubahan hanya pada sumbu X, menghasilkan nilai error 0,06 cm.

Pengujian *inverse kinematics* variasi sumbu Y berisi tentang pengujian metode *inverse kinematics* dengan variasi perubahan hanya pada sumbu Y, menghasilkan nilai error -0,06 cm.

Pengujian *inverse kinematics* variasi sumbu Z berisi tentang pengujian metode *inverse kinematics* dengan variasi perubahan hanya pada sumbu Z, menghasilkan nilai error 0,28 cm.

3.3. Pengujian Sudut Sensor MPU6050

Pengujian Sensor MPU6050 dilakukan dengan cara memberikan variasi nilai sudut pada sudut orientasi *roll* dan *pitch* robot *quadruped*.

3.3.1. Pengujian Sudut Pitch

Pengujian menggunakan variasi sudut *pitch* 0°, 15°, dan -15°, untuk sebagai pembandingan sudut menggunakan busur yang diletakkan disamping bidang. Nilai hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil pengujian sudut pitch

No	Pengujian (derajat)		
	0°	15°	-15°
1	0,33919	17,97343	-17,58513
2	0,33761	17,95347	-17,58549
3	0,32418	17,95212	-17,58062
4	0,31956	17,93516	-17,57164
5	0,31350	17,95847	-17,59668
Rata - rata	0,32681	17,95453	-17,58391

3.3.2. Pengujian Sudut Roll

Pengujian menggunakan variasi sudut *roll* 0°, 15°, dan -15°, untuk sebagai pembandingan sudut menggunakan busur yang diletakkan disamping bidang. Nilai hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian sudut roll

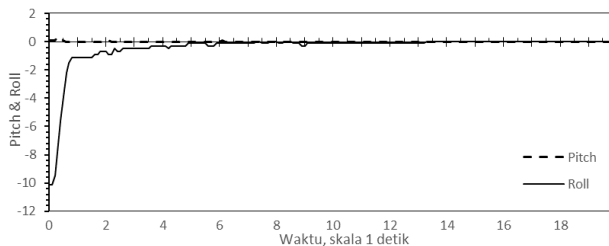
No	Pengujian (derajat)		
	0°	15°	-15°
1	0,17036	15,40485	-14,90678
2	0,16664	15,39956	-14,90141
3	0,14926	15,42812	-14,90416
4	0,15076	15,42233	-14,91016
5	0,15346	15,42101	-14,91086
Rata - rata	0,15809	15,41517	-14,90667

3.4. Pengujian Stabilization Algorithm Robot Quadruped

Pengujian *stabilization algorithm* robot *quadruped* dilakukan dua bagian yaitu melakukan pengujian nilai parameter dan pengujian sudut maksimal dari kestabilan algoritma yang telah dirancang. Pengujian nilai parameter Pengujian sudut maksimal yaitu melakukan pengujian dengan menguji posisi robot *quadruped* pada sudut maksimalnya dapat mempertahankan kestabilan. Berikut pengujian tersebut.

3.4.1. Pengujian Nilai Parameter Stabilization Algorithm

Dalam melakukan pengujian untuk mencari nilai parameter *stabilization algorithm* dilakukan dengan mengubah nilai K_p yang berbeda-beda yang di ambil secara *try and error* dan melihat respon sistem yang dihasilkan. Robot *quadruped* diletakan pada posisi *roll* -10° . Pengujian terbaik dengan variasi nilai parameter $K_p=0.035$ menghasilkan hasil pengujian pada Gambar 11.



Gambar 11. Grafik pengujian nilai parameter stabilization algorithm dengan $k_p=0.035$

Gambar 11. menunjukkan bahwa robot *quadruped* cukup cepat dalam menyetabilkan *body* dengan waktu 5 detik untuk mencapai keadaan *steady state*.

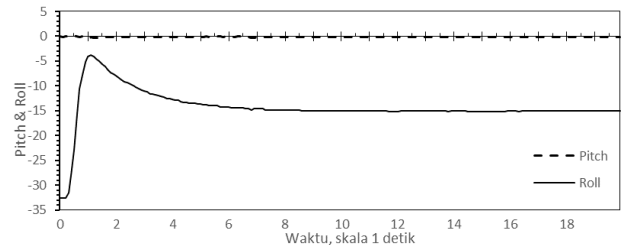
3.4.2. Pengujian Sudut Maksimal Stabilization Algorithm

Dalam melakukan pengujian untuk mencari sudut maksimal yang dapat diproses oleh robot *quadruped* dengan *stabilization algorithm* yaitu dilakukan dengan cara meletakkan robot *quadruped* pada posisi sudut *roll* yang ekstrim. Cara pengujian tersebut dapat dilihat pada Gambar 12.



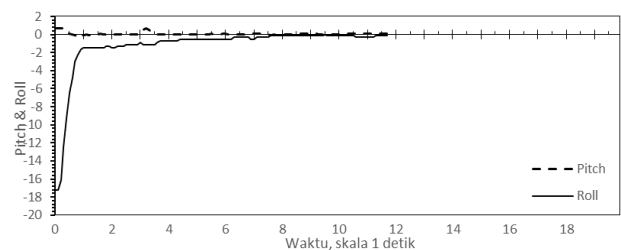
Gambar 12. Pengujian sudut maksimal stabilization algorithm

Pengujian pertama dengan sudut *roll* ekstrim sebesar 33° menghasilkan hasil pengujian dalam berupa grafik sebagai berikut.



Gambar 13. Grafik pengujian sudut maksimal stabilization algorithm dengan sudut 33°

Gambar 13 menunjukkan bahwa robot *quadruped* tidak dapat dalam menyetabilkan *body* dengan dengan sudut *roll* sebesar 33° dan mengalami kemiringan pada sudut *roll* sebesar 15° . Pengujian selanjutnya dengan sudut *roll* ekstrim sebesar 17° menghasilkan hasil pengujian dalam berupa grafik sebagai berikut.



Gambar 14. Grafik pengujian sudut maksimal stabilization algorithm dengan sudut 17°

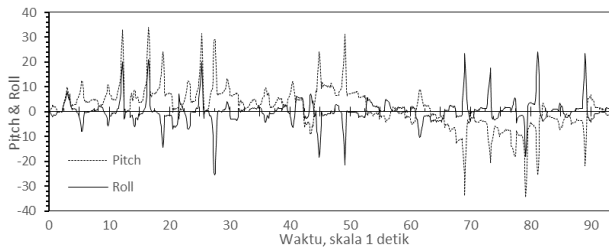
Gambar 14 menunjukkan bahwa robot *quadruped* dapat dalam menyetabilkan *body* dengan dengan sudut *roll* sebesar 17° dan tidak mengalami kemiringan pada sudut *roll*.

3.5. Pengujian Gait Robot Quadruped Pada Uneven Floor

Pengujian *gait* robot *quadruped* dilakukan di track yang dibuat yaitu pada *uneven floor*. Pengujian *gait* robot *quadruped* memiliki 3 bagian pengujian yaitu pengujian *gait* tanpa kontroler dan pengujian *gait* dengan *stabilization algorithm*.

3.5.1. Pengujian Gait Tanpa Kontroler

Setelah dilakukan pengujian *gait* tanpa kontroler maka diperoleh sebuah grafik *pitch* dan *roll* seperti Gambar 15.



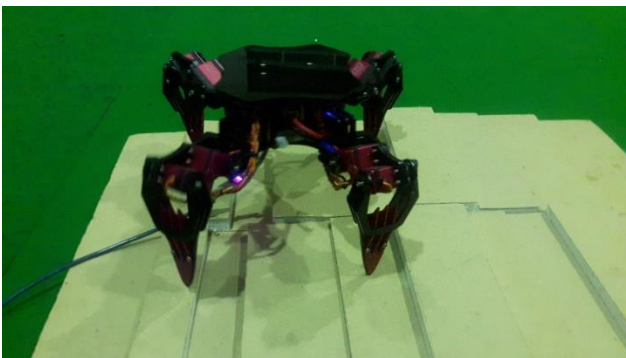
Gambar 15. Grafik pengujian gait tanpa kontroler

Dapat dilihat pada Gambar 15 grafik menunjukkan bahwa dalam 94 detik telah berhasil melewati *uneven floor*. Namun kestabilan robot *quadruped* saat berjalan pada *uneven floor* tanpa kontroler sangatlah tidak bagus karena telah mengalami kemiringan sudut *pitch* maupun sudut *roll* dengan nilai sudut kemiringan terbesar masing-masing sebesar 33.9° untuk sudut *pitch* dan 25.3° untuk sudut *roll*.

3.5.2. Pengujian Gait Dengan Stabilization Algorithm

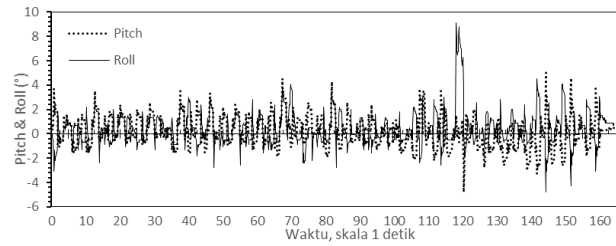
Dapat dilihat pada Gambar 16 yang merupakan pengujian robot *quadruped* dengan *stabilization algorithm* pada saat melewati *uneven floor*. Pada gambar tersebut terlihat jelas bahwa posisi *body* robot *quadruped* tetap stabil walaupun berbeda ketinggian lantainya (*uneven floor*).

Pada pengujian ini dilakukan lima kali pengujian. Pengujian pertama dengan variasi langkah 5 cm. Pengujian kedua dengan variasi langkah 6 cm. Pengujian ketiga dengan variasi langkah 7 cm. Pengujian keempat dengan variasi beban 500 g. Pengujian kelima dengan variasi beban 1000 g. Pengujian robot *quadruped* ini akan melewati lintasan *uneven floor* sepanjang 70 cm.



Gambar 16. Pengujian gait robot quadruped dengan stabilization algorithm

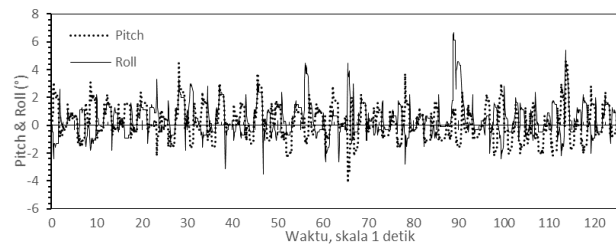
Setelah dilakukan pengujian pertama *gait stabilization algorithm* dengan variasi langkah 5 cm, maka diperolehlah sebuah grafik *pitch* dan *roll* seperti Gambar 17.



Gambar 17. Grafik pengujian gait dengan stabilization algorithm variasi langkah 5 cm.

Dapat dilihat pada Gambar 17, grafik menunjukkan bahwa dalam 160 detik robot *quadruped* telah melewati *uneven floor*. Waktu pengujian yang diperlukan untuk melewati *uneven floor* dengan *stabilization algorithm* memang lebih lama dari pengujian tanpa kontroler, karena *stabilization algorithm* memerlukan waktu beberapa detik untuk melakukan algoritmanya. Hasil kestabilan robot *quadruped* saat berjalan pada *uneven floor* dengan *stabilization algorithm* sangatlah berbeda dengan pengujian *gait* tanpa kontroler. Hal tersebut karena robot *quadruped* saat berjalan pada *uneven floor* memerlukan waktu untuk *stabilization algorithm* setiap kali melangkahkan kaki satu persatu. Dari hasil pengujian tersebut mengalami kemiringan sudut *pitch* maupun sudut *roll* dengan nilai sudut kemiringan terbesar masing-masing sebesar $8,8^\circ$ untuk sudut *pitch* dan $4,8^\circ$ untuk sudut *roll*.

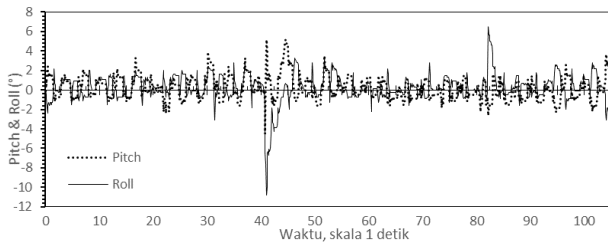
Setelah dilakukan pengujian kedua *gait stabilization algorithm* dengan variasi langkah 6 cm, maka diperolehlah sebuah grafik *pitch* dan *roll* seperti Gambar 18.



Gambar 18. Grafik pengujian gait dengan stabilization algorithm variasi langkah 6 cm.

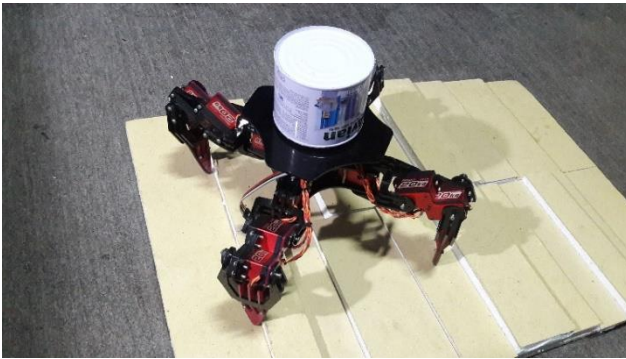
Dapat dilihat pada Gambar 18, grafik menunjukkan bahwa dalam 125 detik robot *quadruped* telah melewati *uneven floor*. Dari hasil pengujian tersebut mengalami kemiringan sudut *pitch* maupun sudut *roll* dengan nilai sudut kemiringan terbesar masing-masing sebesar 6.5° untuk sudut *pitch* dan 4.1° untuk sudut *roll*.

Setelah dilakukan pengujian ketiga *gait stabilization algorithm* dengan variasi langkah 7 cm, maka diperolehlah sebuah grafik *pitch* dan *roll* seperti Gambar 19.



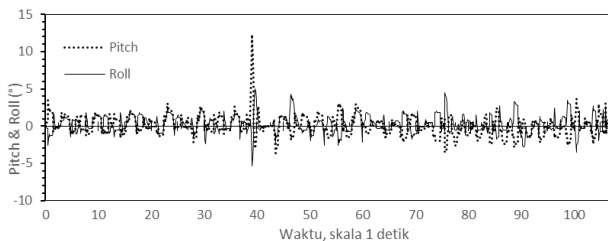
Gambar 19. Grafik pengujian gait dengan stabilization algorithm variasi langkah 7 cm.

Dapat dilihat pada Gambar 19, grafik menunjukkan bahwa dalam 105 detik robot *quadruped* telah melewati *uneven floor*. Dari hasil pengujian tersebut mengalami kemiringan sudut *pitch* maupun sudut *roll* dengan nilai sudut kemiringan terbesar masing-masing sebesar 6.5° untuk sudut *pitch* dan 10.8° untuk sudut *roll*. Pada pengujian keempat ini dapat dilihat seperti pada Gambar 20.



Gambar 20. Pengujian gait robot *quadruped* dengan stabilization algorithm variasi beban 500 g.

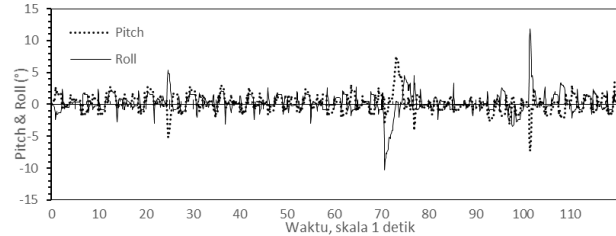
Setelah dilakukan pengujian keempat *gait stabilization algorithm* dengan variasi beban 500g, maka diperoleh sebuah grafik *pitch* dan *roll* seperti Gambar 21.



Gambar 21. Grafik pengujian gait dengan stabilization algorithm variasi beban 500 g.

Dapat dilihat pada Gambar 21, grafik menunjukkan bahwa dalam 105 detik robot *quadruped* telah melewati *uneven floor*. Dari hasil pengujian tersebut mengalami kemiringan sudut *pitch* maupun sudut *roll* dengan nilai sudut kemiringan terbesar masing-masing sebesar 12.3° untuk sudut *pitch* dan 5.4° untuk sudut *roll*.

Setelah dilakukan pengujian kelima *gait stabilization algorithm* dengan variasi beban 1000g, maka diperoleh sebuah grafik *pitch* dan *roll* seperti Gambar 22.



Gambar 22. Grafik pengujian gait dengan stabilization algorithm variasi beban 1000 g..

Dapat dilihat pada Gambar 22, grafik menunjukkan bahwa dalam 115 detik robot *quadruped* telah melewati *uneven floor*. Dari hasil pengujian tersebut mengalami kemiringan sudut *pitch* maupun sudut *roll* dengan nilai sudut kemiringan terbesar masing-masing sebesar 7.3° untuk sudut *pitch* dan 11.4° untuk sudut *roll*. Pada pengujian kelima ini dapat dilihat seperti pada Gambar 23.



Gambar 23. Pengujian gait robot *quadruped* dengan stabilization algorithm variasi beban 1000 g.

Dikarenakan kaki robot *quadruped* yang sudah berpijak pada *uneven floor*, namun tergelincir oleh sebab kaki yang berpijak pada ujung permukaan lantai yang berbeda ketinggian permukaannya, maka robot *quadruped* melakukan *stabilization algorithm* lebih lama dan itu menyebabkan kemiringan sudut maksimal yang lebih besar. Kemudian dari kelima hasil pengujian tersebut dapat dirangkum dalam Tabel 3.

Tabel 3. Pengujian gait dengan stabilization algorithm.

No	Variasi	Waktu	Error sudut <i>pitch</i> terbesar ($^\circ$)	Error sudut <i>roll</i> terbesar ($^\circ$)
1	Langkah 5 cm	160	8,8	4,8
2	Langkah 6 cm	125	6,5	4,1
3	Langkah 7 cm	105	6,5	10,8
4	Beban 500 g	105	12,3	5,4
5	Beban 1000 g	115	7,3	11,4
	Rata-rata	122	8,28	7,3

Setelah dilakukan lima kali pengujian pada robot *quadruped* dengan *stabilization algorithm* pada *uneven floor* maka didapatkan hasil penempuhan waktu 122 detik untuk melewati lintasan sepanjang 70 cm. Menghasilkan *error* sudut *pitch* terbesar yaitu 8,28° dan menghasilkan *error* sudut *roll* terbesar yaitu 7,3°.

Apabila dibandingkan dengan pengujian *gait* tanpa kontroler dibandingkan dengan *gait* yang menggunakan *stabilization algorithm* dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan pengujian gait tanpa kontroler dan gait stabilization algorithm.

No	Gait	Waktu	Error sudut <i>pitch</i> terbesar (°)	Error sudut <i>roll</i> terbesar (°)
1	Gait tanpa kontroler	94	33,9	25,3
2	Gait <i>stabilization algorithm</i>	122	8,28	7,3
	Selisih	-28	25,62	18
	Presentase (%)	-29,8	75,58	71,15

Dengan ini menggunakan *stabilization algorithm* dapat menurunkan waktu tempuh sebesar -29,8 %, akan tetapi dengan rancangan algoritma stabilisasi yang telah dirancang pada penelitian robot *quadruped* ini dapat meningkatkan kestabilan sudut *pitch* sebesar 75,58 % dan kestabilan sudut *roll* sebesar 71,15 %.

4. Kesimpulan

Hasil utama dari penelitian ini telah berhasil mendapatkan kesimpulan bahwa robot *quadruped* dengan menggunakan *stabilization algorithm* dapat meningkatkan kestabilan sudut *pitch* sebesar 75,58 % dan kestabilan sudut *roll* sebesar 71,15 %. Pada pengujian *gait* robot *quadruped* dengan menggunakan *stabilization algorithm* menghasilkan rata-rata kemiringan terbesar sudut *pitch* sebesar 8,28° dan rata-rata kemiringan terbesar sudut *roll* sebesar 7,3. Pada pengujian *gait* robot *quadruped* tanpa kontroler menghasilkan kemiringan terbesar sudut *pitch* sebesar 33,9° dan kemiringan terbesar sudut *roll* sebesar 25,3. Pada hasil pengujian *gait* robot *quadruped* dengan menggunakan *stabilization algorithm* akan menurunkan waktu tempuh sebesar 29,8% dari *gait* tanpa kontroler. Pada hasil pengujian robot *quadruped* dengan *stabilization algorithm* saat diam ditempat dan kemiringan sudut 10° didapatkan hasil nilai parapeter K_p yang terbaik yaitu 0,035. Pada hasil pengujian robot *quadruped* dengan *stabilization algorithm* saat diam ditempat dapat merespon kemiringan sudut maksimal 17° dengan *body* tetap datar. Pada pengujian servo Power HD LF-20 didapatkan rumus konversi, Nilai PWM = 837 + (Sudut x 24.03). Dengan pengujian selisih antara hasil sudut bentukan servo dari hasil rumus konversi dan pengukuran busur derajat, menghasilkan rata-rata *error* sudut servo 0.17°. Pada pengujian inverse kinematic untuk sumbu x memiliki *error* sebesar 0,06, untuk sumbu y memiliki *error* -0,06, dan untuk sumbu z memiliki *error* 0,28. Pada sumbu orientasi

pitch terjadi *error* 0,32681° pada saat pengujian 0°, *error* 2,95453° pada saat pengujian 15°, dan *error* -2,58391° pada saat pengujian -15°. Pada sumbu orientasi *roll* terjadi *error* 0,15809° pada saat pengujian 0°, *error* 0,41517° pada saat pengujian 15°, dan *error* 0,09333° pada saat pengujian -15°.

Referensi

- [1]. Hidayat, "Design And Implementation Inverse Kinematics And Sine Pattern Methods For Locomotion Control On Autonomous Quadruped Robot," e-Proceeding of Engineering, vol. 3, no. 2, 2016.
- [2]. M. Asrofi, "Perancangan Robot Berkaki 6 dalam Mempertahankan Bodi Robot pada Keadaan Datar Menggunakan 9-DOF IMU berbasis Invers Kinematic," Transient, vol. 4, no. 1, 2015..
- [3]. D. Y. Habibi, "Penerapan Inverse Kinematic Pada Pengendalian Gerak Robot," ITS-Undergraduate--Paper, vol. 19303, no. 1808760.
- [4]. R. Nurfansyah, "Estimasi Sudut Orientasi Benda Menggunakan Sensor 6 DOF IMU dan Sensor Magnetometer 3 Aksis," Transient, vol. 2, no. 3, 2013.
- [5]. I. Setiawan, Kontrol PID untuk Proses Industri, Jakarta: Elex Media Komputindo, 2008.
- [6]. A. Riko, "Perancangan Robot Bipedal dengan Sistem Berjalan berbasis Inverse Kinematic dengan Sensor Mpu 6050 Sebagai Indikator Kemiringan," Transient, vol. 6, no. 1, 2017.
- [7]. I. Inc., "MPU-6000 and MPU-6050 Product Specification," Inven. Inc, vol. 1, no. 408, pp. 1-57, 2013.
- [8]. A. Syaifudin, "Perancangan Sistem Kendali Pergerakan Arm Manipulator Berbasis Sensor Inertial Measurement Unit (Imu) dan Sensor Flex," Transient, vol. 1, no. 1, 2017.
- [9]. P. Turner, Mathematics required for Legged Robotic Motion, Tribotix, 2006.