

Pengembangan Algoritma Gait Robot KRSRI ITN Malang Menggunakan Wave Gait dan Analisis Perbandingan Kinerjanya dengan Tripod Gait

Joni Zakariya^a, Radimas Putra Muhammad Dafi Labib^b, Aryuanto Soetedjo^c

^{abc}Department of Electrical Engineering, Institut Teknologi Nasional Malang, Indonesia

ARTICLE INFO

Article history:

Received 12 April 2025
Received in revised form
10 Mei 2025
Accepted 25 Mei 2025
Available online 15 Juni 2025

Keywords:

Hexapod
KRSRI
Wave Gait
Tripod Gait
Kinematics

ABSTRACT

This research develops a gait algorithm for the KRSRI ITN Malang robot using the wave gait method and compares its performance with the tripod gait in terms of energy efficiency and navigation accuracy on various types of terrain. Tests were conducted on flat terrain and broken roads with speed variations from 10 mm/s to 40 mm/s. Results show that on flat terrain, the wave gait is more energy efficient at low speeds (10-13 mm/s) with an average consumption of 0.338 Wh to 0.252 Wh compared to the tripod gait which consumes 0.343 Wh to 0.271 Wh. However, at high speeds (≥ 16 mm/s), the tripod gait was more efficient with a consumption of 0.098 Wh at 40 mm/s compared to the wave gait which consumed 0.195 Wh. In terms of distance traveled accuracy on flat terrain, the tripod gait was superior with an average of 91.7% compared to the wave gait which only reached 90.0%. On broken road terrain, the wave gait was superior in navigation accuracy with an average of 83.7% compared to the tripod gait which only reached 80.8%.

1 Pendahuluan

KRSRI (Kontes Robot Sar Indonesia) adalah jenis robot berjalan kaki yang dirancang khusus untuk misi pencarian dan penyelamatan korban bencana, terutama di Indonesia. Inspirasi untuk tema robot ini muncul dari tantangan yang dihadapi setelah terjadinya gempa bumi. KRSRI 2024 bertujuan untuk mengatasi berbagai rintangan dan menyelesaikan tantangan menyelamatkan korban dari lokasi bencana ke zona aman yang telah ditentukan[1].

Rintangan tahun 2024 meliputi empat jenis, yaitu jalan miring, jalan pecah, jalan berpuing, dan jalan berlumpur. Untuk mempermudah dan mempercepat pembuatan rintangan, rintangan jalan berpuing direalisasikan dengan menggunakan batu koral putih berukuran 3-5 cm, yang biasanya digunakan dalam taman. Jalan berlumpur direalisasikan dengan menggunakan kelereng sebagai tantangan bagi kaki robot yang bisa terperosok di antara tumpukan kelereng tersebut. Sedangkan jalan pecah memerlukan upaya memotong papan triplex dengan pola yang sudah ditentukan. Untuk jalan miring, terdapat jalan menurun dan jalan menanjak, dimana jalan menanjak dilengkapi dengan anak tangga untuk membantu pijakan kaki robot[1].

Robot *hexapod* sering digunakan dalam berbagai bidang, seperti pencarian sumber daya, misi penyelamatan setelah bencana, dan keperluan militer. Dengan enam kaki yang dapat bergerak secara independen, robot ini mampu melewati medan yang sulit dan tidak rata, sehingga cocok untuk tugas-tugas di lingkungan yang menantang[2]. Robot ini unggul dalam menjaga stabilitas, baik secara statis maupun dinamis, terutama saat bergerak di medan yang tidak rata[3][4]. Setiap kaki robot *hexapod* dikendalikan menggunakan metode *inverse kinematics* untuk menentukan sudut pada setiap sendi[5]. Untuk mengontrol keseluruhan tubuh robot, diperlukan metode lain, seperti *forward kinematics*, yang digunakan untuk menentukan sudut awal setiap kaki. Ketika robot hendak melangkah, diperlukan *footstep trajectory*[6]. Dalam pembuatan *footstep trajectory*, digunakan berbagai persamaan untuk membentuk lintasan kaki, seperti lintasan horizontal linier atau lintasan parabola agar kaki dapat mengayun[7].

Saat robot ingin berputar di tempat atau berpindah dari posisi awal ke tujuan, metode kontrol dengan persamaan matriks translasi dan rotasi sangat diperlukan[8]. Robot *hexapod* memiliki berbagai pola langkah (*gait*), dengan dua yang paling umum adalah algoritma *tripod gait* dan *wave gait*[9]. Kedua algoritma ini mempunyai model pergerakan yang berbeda secara fisik, seperti algoritma *wave gait* digunakan untuk menggerakkan kaki secara bergantian dalam pola yang menyerupai gelombang, dengan lima kaki tetap ditahan (*phase stance*) pada setiap waktu menjaga stabilitas yang tinggi dan satu kaki melayang di udara (*phase swing*), sedangkan *tripod gait* didasarkan pada pembagian kaki robot menjadi dua kelompok. Selama setiap langkah,

kelompok pertama diangkat, diputar ke depan, dan kemudian diletakkan kembali di tanah. Setelah itu, kelompok kedua diangkat. Kedua kelompok kaki ini bergerak secara bergantian, dimana kelompok pertama bergerak ke belakang, kelompok kedua bergerak ke depan[10]. Algoritma *wave gait* dan *tripod gait* memiliki perbedaan dalam penggunaan energi saat navigasi. Algoritma *wave gait* lebih efisien pada kecepatan rendah, sedangkan *tripod gait* lebih efisien ketika robot bergerak dengan kecepatan tinggi[11][12].

Robot KRSRI ITN Malang adalah robot dengan desain berbasis *hexapod*, yaitu robot berkaki enam. Pada saat ini algoritma yang digunakan adalah *tripod gait*. *Servo* yang digunakan robot ini adalah *dynamixel AX12A*. Masalah yang dihadapi robot saat ini yaitu sulitnya proses navigasi dan borosnya energi pada saat robot berjalan di medan yang bervariasi.

Oleh sebab itu, penulis akan merancang algoritma *wave gait* sekaligus melakukan analisis untuk meningkatkan navigasi yang lebih efektif dan penggunaan energi yang lebih efisien. Penelitian ini dilakukan dengan membandingkan algoritma *wave gait* dengan algoritma *tripod gait* berdasarkan kemampuan robot bergerak di medan datar dan jalan pecah. Penelitian ini diharapkan mampu menemukan algoritma *gait* yang paling sesuai dengan karakteristik medan tersebut, sehingga robot dapat bergerak lebih optimal dengan penggunaan energi yang rendah tanpa mengurangi keseimbangan atau kualitas kinerja.

2 Studi Literatur

2.1 Tinjauan dari Sistem yang ada

Dari artikel jurnal [13], berfokus untuk merancang, membangun, dan mengontrol robot *hexapod*. Selain itu, penelitian ini membahas gaya berjalan dasar yang digunakan oleh robot berkaki untuk bergerak. Hasil dari proyek ini adalah sebuah robot berkaki yang dapat berjalan menggunakan *tripod gait*, *wave*, dan *ripple*, serta dilengkapi dengan sonar, *force-sensitive resistors*, dan *encoder*. Robot ini dikontrol dan dipantau melalui program antarmuka pengguna yang mampu menampilkan data dari sensor serta posisi kaki robot. Namun penelitian ini tidak menganalisis konsumsi energi dan akurasi navigasi pada jenis *gait* robot.

Pada penelitian jurnal [7], menjelaskan bahwa perancangan model kinematika robot *hexapod* dilakukan dengan menggunakan matriks transformasi homogen, *inverse kinematics*, dan analisis ruang kerja ujung kaki. *Trajectory* ujung kaki pada fase ayun dirancang menggunakan fungsi sinus, sedangkan pada fase tumpuan menggunakan fungsi garis lurus horizontal. Namun, penelitian ini hanya berfokus pada *tripod gait* tanpa membahas atau membandingkannya dengan pola *gait* lainnya, seperti *wave gait*. Selain itu, meskipun ada eksperimen fisik, pengujian hanya dilakukan pada lintasan datar dan tidak mencakup medan yang

tidak terstruktur. Analisis efisiensi energi juga kurang diperhatikan, padahal aspek ini sangat penting untuk operasi robot *hexapod*, terutama dalam misi dengan keterbatasan sumber daya energi.

Menurut jurnal [11], algoritma *wave gait* paling efisien secara energi untuk kecepatan rendah dan algoritma *tripod gait* lebih efisien untuk pergerakan yang lebih cepat. Namun dari pebelitian ini, semua percobaan dilakukan di medan datar dengan permukaan yang lembut.

2.2 Tinjauan Lapangan

Berdasarkan survei lapangan saat robot KRSRI ITN Malang mengikuti lomba pada tahun 2023, Ditemukan masalah pada robot, yaitu kurangnya efektivitas dalam proses navigasi dan efektivitas dalam konsumsi energi selama robot beroperasi. Robot ini menggunakan algoritma *tripod gait*, di mana tiga kaki menempel di tanah dan tiga kaki lainnya melayang di udara untuk melangkah. Algoritma *tripod gait* memiliki perbedaan pergerakan secara fisik dibandingkan dengan *wave gait*. Pada *wave gait*, lima kaki tetap berada di tanah dan satu kaki melayang di udara saat melangkah.

Oleh karena itu, penulis merancang algoritma *wave gait* untuk robot KRSRI ITN Malang yang berbasis pada robot *hexapod*, serta melakukan pengujian langsung pada robot tersebut. Tujuan dari penelitian ini adalah membandingkan efisiensi energi dan jarak tempuh navigasi antara algoritma *wave gait* dan *tripod gait* pada medan datar maupun jalan yang tidak rata. Harapannya, penelitian ini dapat memberikan analisis mengenai algoritma yang paling sesuai untuk medan seperti jalan datar dan jalan pecah, sekaligus memberikan kontribusi signifikan dalam pengembangan robot KRSRI ITN Malang.

2.3 Kontes Robot SAR Indonesia (KRSRI)

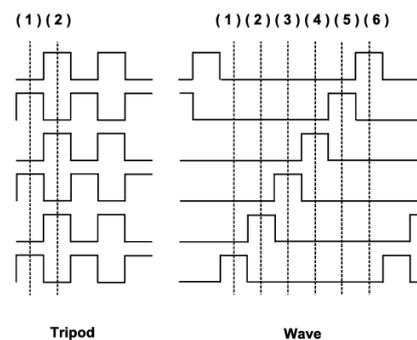
KRSRI (Kontes Robot SAR Indonesia) ini menekankan pada misi pencarian dan penyelamatan bencana yang umum terjadi khususnya di Indonesia. Tema tahun ini sebagaimana tema tahun lalu masih terinspirasi adanya berbagai kesulitan yang terjadi pasca bencana gempa, seperti yang terjadi baru-baru ini di Cianjur akhir November 2022. KRSRI 2024 difokuskan untuk melewati berbagai macam variasi rintangan sebagai ilustrasi kondisi pasca bencana khususnya gempa. Robot juga mendapat tantangan untuk menyelamatkan korban dari lokasi tertentu menuju zona aman (*safety zone*) yang juga sudah ditentukan. Rintangan tahun 2024 terdiri dari 4 jenis rintangan, yaitu: jalan miring, jalan pecah, jalan berpuing, dan jalan berlumpur sebagai ilustrasi kondisi riil pasca bencana khususnya gempa. Untuk memudahkan dan mempercepat peserta dalam membuat rintangan, rintangan jalan berpuing direalisasikan dengan batu koral putih yang biasa digunakan untuk taman berukuran 3-5cm. Adapun jalan berlumpur direalisasikan dengan kelereng sebagai tantangan kaki robot yang bisa terperosok kedalam tumpukan kelereng. Sedangkan jalan pecah sedikit memerlukan upaya memotong papan triplex menggunakan pola yang sudah ditentukan. Untuk jalan miring terdiri dari jalan menurun dan jalan menaik. Jalan menaik diberikan anak tangga untuk membantu pijakan kaki robot. Selain rintangan tersebut, dinding jalur lintasan dibatasi hanya 10cm dari lantai. Berbagai kondisi rintangan tersebut menuntut peserta untuk melengkapi robotnya dengan sistem kendali dan kecerdasan dalam mengolah berbagai sensornya dengan berbagai kendala lintasannya[1].

2.4 Robot Hexapod

Robot *hexapod* merupakan kendaraan mekanik yang menggunakan 6 kaki untuk berjalan, robot ini mempunyai mobilitas yang lebih besar di medan yang tidak teratur dan tidak disengaja, yang tujuannya diperlukan untuk lingkungan yang berbahaya bagi manusia, seperti ladang ranjau, pembangkit listrik tenaga nuklir. Robot *hexapod* memiliki stabilitas statis yang lebih baik saat bergerak dan berdiri[14][15].

2.5 Algoritma Gait Robot

Algoritma *wave gait* dirancang untuk memberikan stabilitas maksimum, di mana hanya satu kaki bergerak (mengayun) pada satu waktu, sementara lima kaki lainnya tetap menapak di tanah untuk menjaga keseimbangan robot. *Wave gait* bekerja dengan cara menggerakkan kaki robot secara berurutan dalam satu siklus langkah penuh. Pola ini cocok untuk navigasi di medan yang tidak rata atau ketika robot membutuhkan kestabilan tinggi. *Tripod gait* adalah pola gerak pada robot berkaki yang menggunakan dua kelompok kaki secara bergantian dalam setiap langkahnya. *Gait* ini memiliki kecepatan tinggi karena setengah dari kaki robot bergerak maju sementara setengah lainnya menopang tubuh. Namun, kestabilannya rendah karena dalam beberapa momen, hanya satu kaki yang menopang setengah dari berat robot, yang dapat menyebabkan selip atau bahkan jatuh[13].



Gambar 1. Algoritma *wave* dan *tripod gait* [13]

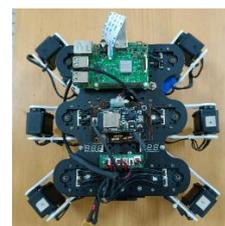
Gambar 1 merupakan diagram yang menunjukkan pola pergerakan kaki robot dalam suatu periode waktu, di mana posisi tinggi menandakan kaki terangkat dan posisi rendah menandakan kaki menyentuh tanah. Dalam *tripod gait*, kaki terbagi dalam dua kelompok, dan setiap kelompok bergerak bersamaan, sehingga robot dapat bergerak lebih cepat namun kurang stabil. Sementara itu, dalam *wave gait*, hanya satu kaki yang bergerak pada satu waktu, memberikan stabilitas tinggi tetapi kecepatan lebih lambat karena setiap langkah harus diselesaikan satu per satu sebelum kaki lainnya bergerak.

3 Metodologi Penelitian

3.1 Waktu dan Tempat Pelaksanaan

Penelitian ini dilakukan di dua tempat yaitu Laboratorium Otomasi Teknik Elektro untuk perancangan algoritma robot, dan di dalam Laboratorium Robotika untuk pengambilan data konsumsi energi dan efektivitas navigasi di medan jalan datar dan jalan pecah selama robot beroperasi.

3.2 Spesifikasi Robot KRSRI



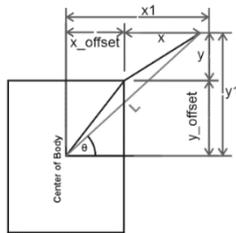
Gambar 2. Robot KRSRI ITN Malang

Gambar 2 merupakan robot KRSRI ITN Malang yang mempunyai enam kaki, di mana setiap kaki terdiri dari tiga sendi yang digerakkan oleh aktuator *servo dynamixel AX-12A*. Sistem kendali menggunakan mikrokontroler *ESP32 WROOM*, sementara sumber daya utamanya berasal dari baterai *Li-Po 2200 mAh 3S 12V*. Struktur rangka robot terbuat dari akrilik dengan

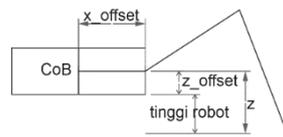
dimensi 18.5 x 15.5 x 8.5 cm. Robot ini dilengkapi fitur pemantauan arus, tegangan, daya, serta konsumsi energi yang dapat diakses melalui *web* server dalam jaringan lokal. Selain itu, kecepatan robot juga dapat diatur melalui *web* server, memungkinkan kontrol yang lebih fleksibel dan efisien.

3.3 Kinematika Robot

Body kinematics digunakan untuk mengubah koordinat kaki berdasarkan posisi dan orientasi tubuh robot. Untuk memahami proses ini, diperlukan ilustrasi yang menunjukkan hubungan antara tubuh robot dengan kaki-kakinya. Gambar 3 dan 4 memperlihatkan sambungan antara tubuh robot dan kaki-kakinya.



Gambar 3. Tampak atas



Gambar 4. Tampak depan

Keterangan dari Gambar 3 dan 4:

- θ : sudut awal kaki robot berdasarkan orientasi tubuh robot.
- x_offset : pergeseran koordinat x, dari kaki ke tubuh robot.
- y_offset : pergeseran koordinat y, dari kaki ke tubuh robot.
- z_offset : pergeseran coordinate z, dari tinggi robot ke tubuh robot.
- x, y, z : Posisi kaki dalam koordinat berdasarkan sendi yang terhubung ke tubuh robot.
- $x1, y1$: Posisi proyeksi kaki dalam koordinat tubuh robot (*body frame*).
- *tinggi robot* : tinggi robot dari tanah.
- L : Panjang kaki dari pusat tubuh robot ke ujung kaki.

Ilustrasi di atas merupakan contoh dari 1 kaki robot, untuk menentukan semua kaki robot, maka variable di atas diberi tambahan variable i , dimana i adalah $\{R1, R2, R3, L1, L2, L3\}$ kaki robot. Untuk menyelesaikan persamaan di atas, disini penulis menggunakan metode forward kinematics. Berikut bentuk persamaannya, merujuk pada Persamaan (1) dan (2).

$$x1_i = \cos(\theta_i) * L \quad (1)$$

$$y1_i = \sin(\theta_i) * L \quad (2)$$

Pergeseran *Offset* x_i, y_i, z_i merujuk pada Persamaan (3), (4) dan (5):

$$x_i = x1_i - x_offset_i \quad (3)$$

$$y_i = y1_i - y_offset_i \quad (4)$$

$$z_i = \text{tinggi robot} + z_offset_i \quad (5)$$

Dengan diketahui $P_{body,i}(x_i, y_i, z_i)$, posisi kaki robot yang terhubung dengan body sudah bisa di operasikan dengan matriks rotasi dan translasi menggunakan transformasi homogen. Transformasi homogen digunakan untuk melakukan operasi matriks rotasi dan translasi dengan koordinat kaki robot secara langsung tanpa harus melakukan operasi secara terpisah. Berikut bentuk persamaan dari matriks rotasi dan translasi merujuk pada Persamaan (6), (7), (8), (9), dan (10).

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (6)$$

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (7)$$

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Kombinasi matriks rotasi dari Persamaan (6), (7), dan (8) secara berurutan merujuk pada Persamaan (9).

$$Rot_{xyz} = R_x \cdot R_y \cdot R_z \quad (9)$$

Persamaan matriks translasi merujuk pada Persamaan (10).

$$Trans_{xyz} = \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} \quad (10)$$

Untuk melakukan operasi jadi satu matriks translasi dengan rotasi, maka dibutuhkan transformasi homogen dengan dimensi 4x4 pada matriks rotasi dan translasi. Berikut bentuk persamaannya merujuk pada Persamaan (11), (12), (13), (14), (15), dan (16).

$$P_{body,i} = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (11)$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$Trans_{xyz} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Sehingga persamaan menjadi seperti Persamaan (16):

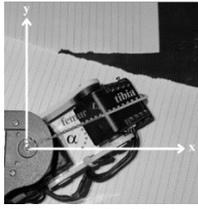
$$T_{xyz,i} = \begin{bmatrix} Rot_{xyz} & Trans_{xyz} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{body,i} \\ 1 \end{bmatrix} \quad (16)$$

Keterangan:

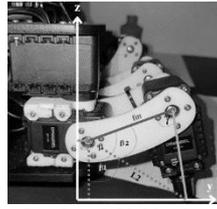
- ϕ, θ, ψ : sudut rotasi tubuh robot.
- tx, ty, tz : offset translasi dari tubuh robot.
- $P_{body,i}$: posisi awal kaki dalam kerangka tubuh robot (x_i, y_i, z_i).
- Rot_{xyz} : matriks rotasi dari sumbu x,y,z
- $Trans_{xyz}$: matriks translasi untuk koordinat x,y,z
- $T_{xyz,i}$: hasil operasi transformasi homogen dari Rot_{xyz} , $Trans_{xyz}$, dan $P_{body,i}$.

Setelah $T_{xyz,i}$ didapatkan, maka nilai tersebut sudah bisa dimasukkan ke dalam inverse kinematics pada tiap kaki robot.

Pada implementasi ini, *inverse kinematics* digunakan untuk menghitung masing-masing kaki robot yang menggunakan 3 *DoF* (*Degree of Freedom*), persamaan *inverse kinematics* dapat dirumuskan dengan melihat sambungan dalam satu arah gerakan. Gambar 5 dan 6 menunjukkan tampilan kaki robot tampak atas dan depan.



Gambar 5. Kaki robot tampak atas



Gambar 6. Kaki robot tampak depan

Dari Gambar 5 dan 6, dapat kita buat persamaannya dengan menggunakan dasar trigonometri, berikut bentuk persamaan merujuk pada Persamaan (17), (18), (19), (20), (21), (22), dan (23).

$$\alpha = \text{atan} \frac{y}{x} \quad (17)$$

$$L1 = \sqrt{x^2 + y^2} \quad (18)$$

$$L2 = \sqrt{(L1 - cx)^2 + y^2} \quad (19)$$

$$\beta2 = \text{acos} \left(\frac{\sqrt{fm} + \sqrt{L2} - \sqrt{L1}}{2 fm L2} \right) \quad (20)$$

$$\beta1 = \text{acos} \frac{z}{L2} \quad (21)$$

$$\beta = \beta1 + \beta2 \quad (22)$$

$$\gamma = \text{acos} \left(\frac{\sqrt{fm} + \sqrt{L1} - \sqrt{L2}}{2 fm L1} \right) \quad (23)$$

Keterangan:

- α : sudut servo alpha.
- β : sudut servo beta.
- γ : sudut servo gamma.

3.4 Footstep Trajectory Planning

Footstep Trajectory planning adalah proses merencanakan jalur atau lintasan yang digunakan sebagai referensi langkah pada robot *hexapod*. Dalam perencanaan lintasan ini, digunakan konsep dasar trigonometri, seperti fungsi sinus (\sin) dan mengambil dasar persamaan dari interpolasi linier, untuk menentukan gerakan kaki robot. Untuk menentukan durasi *trajectory*, di sini peneliti membuat *trajectory* yang bergantung pada waktu, dimana dalam satu periode waktunya dapat dicari dengan menggunakan rumus kecepatan. Berikut adalah persamaan matematis dari interpolasi linear merujuk pada Persamaan (24) dan (25).

Persamaan untuk x merujuk pada Persamaan (24):

$$x = x_1 + t \times (x_2 - x_1) \quad (24)$$

Persamaan untuk y merujuk pada Persamaan (25):

$$y = y_1 + t \times (y_2 - y_1) \quad (25)$$

Selain interpolasi linear untuk x dan y, lintasan pada sumbu z dalam penelitian ini menggunakan fungsi sinusoidal. Hal ini bertujuan untuk menghasilkan lintasan berebentuk parabola. Bentuk persamaan untuk z merujuk pada Persamaan (26).

$$z = z_{max} + \sin \times (\pi \cdot t) \quad (26)$$

Keterangan:

- z: posisi interpolasi pada sumbu z.
- z_{max} : Ketinggian maksimum yang dicapai robot pada lintasan sumbu z.
- π : Konstanta pi ($\pi \approx 3.14$).
- t: parameter posisi terhadap waktu dengan nilai $0 < t < 1$.

Setelah mengetahui semua persamaan untuk melakukan *trajectory planning*, penulis mengatur perubahan nilai t berdasarkan kecepatan yang ditentukan. Berikut bentuk persamaan dari kecepatan merujuk pada Persamaan (27).

$$W = \frac{J}{K} \quad (27)$$

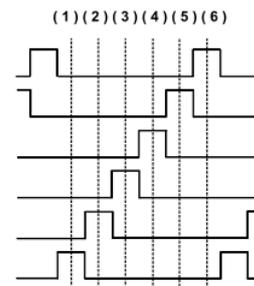
Keterangan:

- J: jarak translasi robot.
- K: kecepatan robot dalam mm/s.
- W: waktu/durasi robot dalam melakukan *trajectory*.

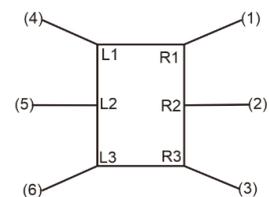
Jadi perubahan untuk nilai t dari 0 sampai 1 akan bergantung pada variabel W.

3.5 Algoritma Wave Gait

Algoritma *wave gait* digunakan untuk menggerakkan kaki secara bergantian dalam pola yang menyerupai gelombang, dengan lima kaki tetap ditanah (*phase stance*) pada setiap waktu menjaga stabilitas yang tinggi dan satu kaki melayang di udara (*phase swing*). Gambar 7 adalah diagram dari algoritma wave gait.



Gambar 7. Diagram waktu wave gait



Gambar 8. Identitas kaki

Pada Gambar 7 hanya 1 kaki berada dalam *phase swing* pada satu waktu, sedangkan 5 kaki lainnya berada dalam *phase stance*. Dengan begitu, robot bisa melakukan translasi maupun rotasi secara dinamis. Gambar 8 menjelaskan hubungan kaki mana yang harus mengalami *phase swing* maupun *phase stance*.

3.6 Konsumsi Energi

Untuk mengukur konsumsi energi selama robot beroperasi, diperlukan beberapa sensor, seperti sensor arus dan tegangan untuk mendapatkan daya. Kemudian dengan menghubungkan daya dengan waktu, maka didapatkan konsumsi energy robot. Berikut bentuk persamaannya merujuk pada Persamaan (28).

$$E = \sum_0^t (I * V * dt) / 3600 \quad (28)$$

Keterangan:

- Σ : symbol untuk penjumlahan, karena setiap waktu energy tidak selalu tetap.
- t: waktu total yang ingin dihitung, dibagi menjadi banyak bagian kecil (dt).
- dt: selang waktu (s).
- V: Tegangan (V).
- I: Arus (A).
- 3600: Konversi dari detik ke jam.

4 Hasil dan Pembahasan

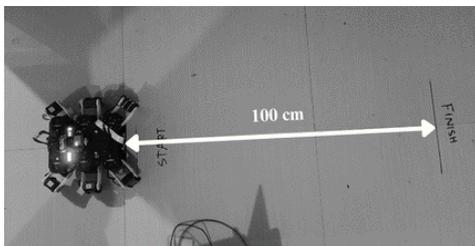
4.1 Proses Pengujian Algoritma Wave Gait dan Tripod Gait

Pada pengujian ini, robot dijalankan pada jalan datar, dan jalan pecah. Untuk melakukan pengujian kinematika robot, maka diperlukan perubahan kecepatan robot untuk mendapatkan data konsumsi energi, dan akurasi jarak tempuh. Disini penulis melakukan perubahan kecepatan dan mengambil data konsumsi energi melalui *web server*. Berikut merupakan tampilan *web server* yang digunakan dalam penelitian ini.



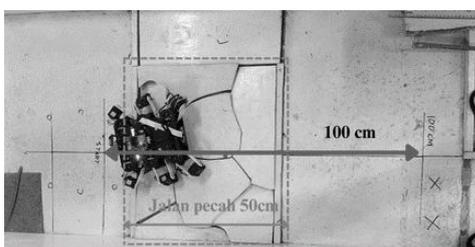
Gambar 9. Tampilan web server

Dari Gambar 9 Penulis dapat merubah kecepatan robot dan memantau *timer* atau durasi selama berjalan, dan konsumsi energi. Kemudian untuk percobaan dilakukan di jalan datar dan jalan pecah.



Gambar 10. Percobaan jalan datar

Gambar 10 merupakan percobaan jalan datar dilakukan dengan cara berjalan lurus mulai dari titik start sampai titik *finish* dengan panjang 100 cm.

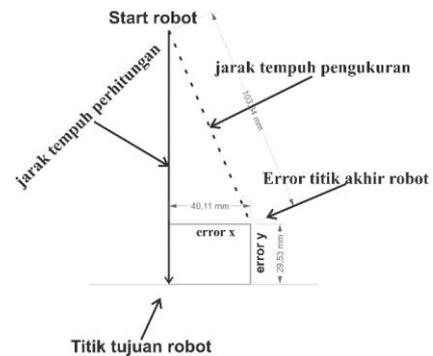


Gambar 11. Percobaan jalan pecah

Gambar 11 merupakan percobaan jalan pecah, pada percobaan ini, robot dijalankan lurus menuju titik *finish* dengan total panjang waktu tempuh 100 cm dan melewati jalan pecah dengan panjang 50 cm.

4.2 Proses Pengukuran Data Jarak Tempuh

Dengan mengukur koordinat tujuan robot menggunakan meteran, maka bisa didapatkan jarak tempuh robot selama melakukan navigasi, Gambar 12 merupakan pengukuran data jarak tempuh robot.



Gambar 12. Pengukuran jarak tempuh robot

Dari Gambar 12 untuk jarak tempuh robot dapat dihitung dengan menggunakan dasar persamaan *pythagoras*, berikut bentuk persamaanya merujuk pada Persamaan (29).

$$L = \sqrt{(L1 - error y)^2 + error x^2} \quad (29)$$

Keterangan:

- L : Jarak tempuh pengukuran.
- $L1$: Jarak tempuh perhitungan.

4.3 Hasil Perancangan Algoritma Wave Gait

Dengan melakukan beberapa metode yang sudah di jelaskan pada bab sebelumnya, algoritma *wave gait* telah berhasil dirancang dan diimplentasikan langsung pada robot KRSRI ITN Malang. Berikut merupakan Tabel 1 robot yang sedang berjalan menggunakan algoritma *wave gait*.

Tabel 1. Hasil perancangan algoritma *wave gait*

Step	Gambar	Kaki	Phase
1		R1	Swing
2		R2	Swing
3		R3	Swing
4		L1	Swing
5		L2	Swing
6		L3	Swing

Dari table 1 terlihat bahwa algoritma *wave gait* telah berhasil di implentasikan pada robot KRSRI ITN Malang. Dimana pada saat step 1, kaki R1 robot akan mengalami *phase swing*, dan

phase swing ini akan bergantian mulai dari step 1 = R1, step 2 = R2, step 3 = R3, step 4 = L1, step 5 = L2, dan step 6 = L3.

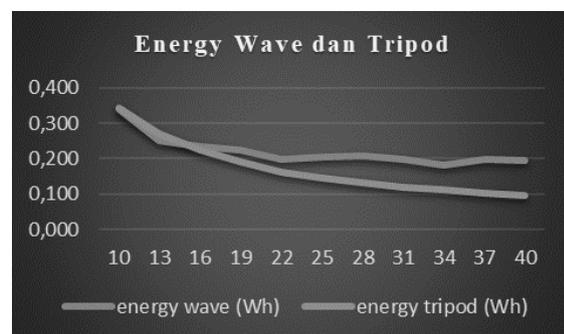
4.4 Analisa Hasil Algoritma Wave Gait dan Tripod Gait

Pada sub bab ini, peneliti akan membahas hasil analisis konsumsi energi, dan akurasi jarak tempuh yang merupakan parameter dari *output* algoritma *wave gait* dan *tripod gait*. *Output* dari kedua algoritma tersebut akan di analisis dan kemudian akan dibandingkan. Pada saat melakukan percobaan untuk mendapatkan parameter konsumsi energi, dan akurasi jarak tempuh, penulis melakukan 3 kali percobaan yang sama pada kecepatan untuk mendapatkan data yang lebih akurat, kemudian di ambil rata-ratanya. Berikut adalah data yang telah didapatkan selama melakukan percobaan.

Pada percobaan ini, konsumsi energi robot akan diukur dengan menggunakan algoritma *wave gait* dan *tripod gait* pada jalan datar dengan permukaan yang rata. Kemudian di analisis dan keduanya akan dibandingkan manakah kecepatan yang tepat dengan konsumsi energi yang efisien.

Tabel 2. Konsumsi energi wave dan tripod jalan datar

Kecepatan (mm/s)	Durasi (s)	Jarak Tempuh (cm)	Energy Wave (Wh)	Energy Tripod (Wh)
10	100	100	0,338	0,343
13	77	100	0,252	0,271
16	63	100	0,235	0,226
19	53	100	0,226	0,191
22	45	100	0,197	0,163
25	40	100	0,205	0,145
28	36	100	0,208	0,134
31	32	100	0,197	0,118
34	29	100	0,183	0,112
37	27	100	0,199	0,102
40	25	100	0,195	0,098



Gambar 13. Konsumsi energy wave dan tripod jalan datar

Dari Tabel 2 dan Gambar 13, dapat dianalisis bahwa konsumsi energi robot akan menurun jika kecepatan robot semakin meningkat. Hal ini bisa terjadi karena robot dijalankan dengan jarak tempuh yang sama dengan kecepatan berbeda, dengan itu waktu durasi robot beroperasi juga mengalami perubahan, semakin kecil kecepatan robot, maka semakin lama robot melakukan perjalanan menuju titik akhir. Selain itu konsumsi energi pada algoritma *wave gait* hampir sama pada saat kecepatan 22 mm/s sampai 40 mm/s. begitu juga dengan konsumsi *energy tripod gait* akan mengalami penurunan jika kecepatan semakin meningkat. Grafik *tripod* hampir sama dengan grafik *wave*. Namun grafik *tripod* lebih linier mengalami

penurunan energi dibandingkan dengan yang *wave*. Pada kedua algoritma yang telah dianalisis, bahwa konsumsi energi *wave gait* lebih efisien pada kecepatan 10 mm/s sampai 13 mm/s, dan *tripod gait* akan lebih efisien jika kecepatan mencapai 16 mm/s sampai 40 mm/s.

Tabel 3. Akurasi jarak tempuh wave dan tripod jalan datar

Kecepatan (mm/s)	Durasi (s)	Jarak Tempuh (cm)	Akurasi Jarak Tempuh (cm)	
			Wave	Tripod
10	100	100	89,8	92,6
13	77	100	90,6	92,2
16	63	100	89,2	92,3
19	53	100	88,8	92,1
22	45	100	86,6	90
25	40	100	89,6	91,2
28	36	100	91,5	92,3
31	32	100	93,8	90,3
34	29	100	90,1	93,3
37	27	100	93,5	91,2
40	25	100	86,1	91,5
Rata - Rata			90,0	91,7

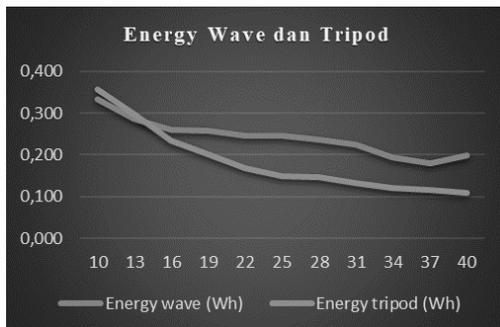


Gambar 14. Akurasi jarak tempuh wave dan tripod jalan datar

Dari Tabel 3 dan Gambar 14, akurasi jarak tempuh tidak ada hubungan langsung pada perubahan kecepatan robot. Hal ini bisa terjadi dari beberapa faktor, seperti gesekan antara kaki dengan permukaan tanah atau dari mekanisme robot yang mempengaruhi kinerja dari kinematic robot. Namun dapat diambil rata-rata pada seluruh percobaan dengan kecepatan dari 10 mm/s – 40 mm/s bahwa algoritma *tripod gait* akurasinya lebih tinggi yaitu 91,7 % dari pada *wave gait* dengan nilai 90,0 % pada saat dijalankan di jalan datar.

Tabel 4. Konsumsi energi wave dan tripod jalan pecah

Kecepatan (mm/s)	Durasi (s)	Jarak tempuh (cm)	Energy wave (Wh)	Energy tripod (Wh)
10	100	100	0,332	0,356
13	77	100	0,287	0,298
16	63	100	0,260	0,233
19	53	100	0,257	0,2
22	45	100	0,246	0,169
25	40	100	0,245	0,15
28	36	100	0,237	0,147
31	32	100	0,224	0,132
34	29	100	0,193	0,121
37	27	100	0,180	0,117
40	25	100	0,198	0,109

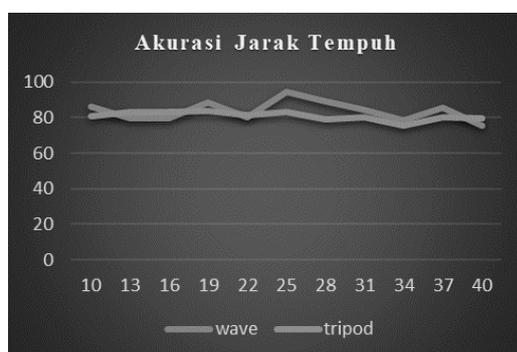


Gambar 15. Konsumsi energi wave dan tripod jalan pecah

Dari Tabel 4 dan Gambar 15 merupakan percobaan jalan pecah, konsumsi energi pada jalan pecah juga sama mengalami penurunan jika kecepatan dinaikkan seperti percobaan jalan datar. Namun konsumsi energi sedikit lebih meningkat pada percobaan jalan pecah. Sama seperti percobaan jalan datar bahwa konsumsi energi tripod lebih linier mengalami penurunan dari pada wave gait. dari grafik di atas algoritma wave gait lebih efisien pada saat kecepatan 10 mm/s – 13 mm/s, jika kecepatan mencapai 16 mm/s sampai 40 mm/s, maka algoritma tripod lebih efisien terhadap konsumsi energi.

Tabel 5. Akurasi jarak tempuh wave dan tripod jalan pecah

Kecepatan	Durasi	Jarak Tempuh	Akurasi Jarak Tempuh (%)	
			Wave	Tripod
10	100	100	85,9	80,7
13	77	100	79,7	82,9
16	63	100	79,6	83,2
19	53	100	88,5	83,7
22	45	100	80	81,1
25	40	100	94,4	83,1
28	36	100	88,9	78,8
31	32	100	84,3	80,2
34	29	100	78,6	75,6
37	27	100	85,7	79,9
40	25	100	75,3	79,6
Rata - Rata			83,7	80,8



Gambar 16. Akurasi jarak tempuh wave dan tripod jalan pecah

Dari Tabel 5 dan Gambar 16, akurasi jarak tempuh pada jalan pecah juga tidak ada hubungan langsung dengan kecepatan. Namun masih bergantung dengan jenis gait yang dipakai, rata – rata akurasi algoritma wave gait mencapai 83,7 %, lebih tinggi dari pada algoritma tripod gait yaitu 80,8 %.

5 Kesimpulan

Penelitian ini bertujuan untuk mengembangkan algoritma gait pada robot KRSRI ITN Malang menggunakan metode wave gait serta menganalisis dan membandingkan kinerjanya dengan metode tripod gait dalam hal efisiensi energi dan akurasi navigasi di berbagai jenis medan. Algoritma wave gait dalam penelitian ini telah berhasil dirancang dan diterapkan pada robot KRSRI ITN Malang. Berdasarkan hasil pengujian dan analisis, diperoleh beberapa kesimpulan penting. Dari segi efisiensi energi, algoritma wave gait lebih hemat daya saat digunakan pada kecepatan rendah (10-13 mm/s), sedangkan algoritma tripod gait lebih efisien pada kecepatan menengah hingga tinggi (16-40 mm/s). Hal ini menunjukkan bahwa pemilihan algoritma gait yang tepat bergantung pada kebutuhan operasional dan kecepatan yang diinginkan. Sementara itu, dalam hal akurasi jarak tempuh, algoritma tripod gait memiliki akurasi yang lebih tinggi pada medan datar dengan rata-rata sebesar 91,7%, dibandingkan dengan wave gait yang mencapai 90,0%. Namun, pada medan jalan pecah, algoritma wave gait menunjukkan akurasi yang lebih baik, yaitu 83,7%, dibandingkan algoritma tripod gait yang hanya mencapai 80,8%. Hasil penelitian ini menegaskan bahwa pemilihan algoritma gait harus mempertimbangkan kondisi medan serta kebutuhan efisiensi energi dan akurasi navigasi robot.

Kontribusi Penulis

Dalam penelitian ini, penulis, Joni Zakariya bersama dengan Radimas Putra Muhammad Davi Labib dan Aryunto Soetedjo bertanggung jawab atas perancangan penelitian, pengumpulan data, serta analisis utama terhadap hasil yang diperoleh.

Ucapan Terima Kasih

Terima kasih kepada Laboratorium Otomasi Industri & Robotika Program Teknik Elektro S1 ITN Malang sebagai tempat penelitian ini dilakukan.

Referensi

- [1] Pedoman Kontes Robot Indonesia (KRI) Pendidikan Tinggi Tahun 2024.
- [2] B. You, Y. Fan, and D. Liu, "Fault-tolerant motion planning for a hexapod robot with single-leg failure using a foot force control method," *International Journal of Advanced Robotic Systems*, vol. 19, no. 5, p. 172988062211210, Sep. 2022, doi: 10.1177/17298806221121070.
- [3] S. Muslimin and D. Istardi, "Inverse Kinematics Analysis for Motion Prediction of a Hexapod Robot," in *2018 International Conference on Applied Engineering (ICAE)*, Batam: IEEE, Oct. 2018, pp. 1–5. doi: 10.1109/INCAE.2018.8579402.
- [4] H. Suprihono and Mashoedah, "Development of Inverse Kinematics Learning Media Using Hexapod Robot for Robotics Course," *J. Phys.: Conf. Ser.*, vol. 1737, no. 1, p. 012035, Jan. 2021, doi: 10.1088/1742-6596/1737/1/012035.
- [5] J. A. Lizarraga, J. A. Garnica, J. Ruiz-Leon, G. Munoz-Gomez, and A. Y. Alanis, "Advances in the Kinematics of Hexapod Robots: An Innovative Approach to Inverse Kinematics and Omnidirectional Movement," *Applied Sciences*, vol. 14, no. 18, p. 8171, Sep. 2024, doi: 10.3390/app14188171.
- [6] H. Xia, X. Zhang, and H. Zhang, "A New Foot Trajectory Planning Method for Legged Robots and Its Application in Hexapod Robots," *Applied Sciences*, vol. 11, no. 19, p. 9217, Oct. 2021, doi: 10.3390/app11199217.
- [7] J. Sun, J. Ren, Y. Jin, B. Wang, and D. Chen, "Hexapod robot kinematics modeling and tripod gait design based on the foot end trajectory," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau: IEEE, Dec. 2017, pp. 2611–2616. doi: 10.1109/ROBIO.2017.8324813.
- [8] L. Fuček, Z. Kovačić, and S. Bogdan, "Analytically founded yaw control algorithm for walking on uneven terrain applied to a hexapod robot," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, p. 1729881419857997, May 2019, doi: 10.1177/1729881419857997.

- [9] N. Namura and H. Nakao, "A Central Pattern Generator Network for Simple Control of Gait Transitions in Hexapod Robots based on Phase Reduction," Apr. 26, 2024, *arXiv: arXiv:2404.17139*. doi: 10.48550/arXiv.2404.17139.
- [10] A. Da Fonseca Braga, E. Barros Costa, L. Rocha Olivi, and A. Silva Barbosa, "Development and Path Planning of a Low Cost Hexapod Mobile Robot," in *Anais do 14^o Simpósio Brasileiro de Automação Inteligente*, Galoa, 2019. doi: 10.17648/sbai-2019-111371.
- [11] M. Luneckas *et al.*, "Hexapod Robot Gait Switching for Energy Consumption and Cost of Transport Management Using Heuristic Algorithms," *Applied Sciences*, vol. 11, no. 3, p. 1339, Feb. 2021, doi: 10.3390/app11031339.
- [12] G. Wang, L. Ding, H. Gao, Z. Deng, Z. Liu, and H. Yu, "Minimizing the Energy Consumption for a Hexapod Robot Based on Optimal Force Distribution," *IEEE Access*, vol. 8, pp. 5393–5406, 2020, doi: 10.1109/ACCESS.2019.2962527.
- [13] M. Žák, "Design, Construction and Control of Hexapod Walking Robot".
- [14] E. H. Hasnaa and B. Mohammed, "Planning tripod gait of an hexapod robot," in *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, Marrakech: IEEE, Mar. 2017, pp. 163–168. doi: 10.1109/SSD.2017.8166964.
- [15] S. Arrigoni, M. Zangrandi, G. Bianchi, and F. Braghin, "Control of a Hexapod Robot Considering Terrain Interaction," *Robotics*, vol. 13, no. 10, p. 142, Sep. 2024, doi: 10.3390/robotics13100142.