

BAB I PENDAHULUAN

1.1 Latar Belakang

Robotika adalah bidang ilmu yang berkembang pesat seiring dengan kemajuan teknologi. Robot diciptakan dengan tujuan utama untuk membantu manusia dalam menyelesaikan berbagai pekerjaan, terutama yang membutuhkan ketelitian tinggi atau berisiko. Salah satu jenis teknologi robotika yang banyak dimanfaatkan di dunia industri adalah lengan robot atau robot manipulator. Teknologi ini memiliki beberapa keunggulan dalam mendukung pekerjaan manusia, seperti mengurangi biaya produksi, meningkatkan produktivitas, mampu menjalankan pekerjaan berulang, memiliki akurasi dan kecepatan tinggi dalam menyelesaikan tugas, serta bisa diprogram ulang untuk berbagai fungsi yang berbeda (Chairunnas & Pamungkas, 2019).

Teknologi robotika lain yang sedang mengalami perkembangan pesat adalah robot *humanoid*. Robot jenis ini dirancang dengan struktur tubuh dan gerakan yang menyerupai manusia. Dalam dunia robotika, dikenal istilah *DoF* (*Degree of Freedom*) atau derajat kebebasan. Secara umum, *DoF* pada robot berarti setiap gerakan linier atau rotasi yang terjadi sepanjang atau mengelilingi suatu sumbu (*axis*). Analisis kinematika pada robot terbagi menjadi dua jenis, yaitu *forward kinematics* dan *inverse kinematics*. *Forward kinematics* merupakan analisis yang digunakan untuk menentukan koordinat posisi (x, y, z) jika sudut pada setiap sendi (*joint*) telah diketahui. Sementara itu, *inverse kinematics* adalah analisis yang bertujuan untuk menemukan nilai sudut pada masing-masing *joint* berdasarkan posisi koordinat (x, y, z) yang diketahui. Penerapan algoritma *inverse kinematics* memungkinkan robot dengan n -*DoF* untuk menghitung kombinasi sudut yang dibutuhkan agar *end effector* robot mencapai posisi akhir tertentu (Kurniawan & Sumardi, 2017).

Selanjutnya ada beberapa metode yang sering digunakan oleh para peneliti untuk menjaga keseimbangan pada robot *humanoid* saat berjalan, salah satunya adalah metode *Zero Moment Point*. Metode ini menghitung jarak gaya untuk mengurangi efek gaya dari pusat massa robot. Selain menjaga keseimbangan, robot

humanoid juga membutuhkan algoritma untuk berjalan. Dalam proses berjalan, robot bipedal melalui empat fase: *Double Support Phase*, saat robot berdiri dengan kedua kaki; *Pre-Swing Phase*, yaitu fase persiapan sebelum mulai berjalan; *Single Support Phase*, di mana robot bertumpu pada satu kaki; dan *Post-Swing Phase*, fase akhir dalam langkah berjalan (Praditya Handi Setiawan, 2016).

Penelitian ini akan membahas sistem berjalan pada robot *humanoid*. Robot *humanoid* ini dirancang dengan 16 derajat kebebasan, terdiri dari 5 derajat kebebasan pada setiap kaki dan 3 pada setiap tangan yang berfungsi sebagai sendi. Pembahasan akan difokuskan pada analisis *inverse kinematics* yang tersegmentasi untuk menentukan sudut-sudut pada *joint* robot *humanoid* yang diperlukan untuk melakukan suatu gerakan. Analisis *inverse kinematics* dilakukan secara terpisah untuk setiap bagian tubuh robot *humanoid*. Robot *humanoid* ini akan menggerakkan *end-effector* berupa telapak kaki dan tangan menggunakan metode *inverse kinematics*. *End-effector* adalah titik akhir dari gerakan yang dilakukan. Dalam analisis *inverse kinematics* ini menggunakan pendekatan geometris dan Pengujian robot dilakukan dengan memantau kemiringannya saat bergerak jalan. Robot yang digunakan untuk penelitian ialah robot *humanoid* jenis bipedal. Tujuan penelitian ini adalah merancang gain robot *humanoid* yang dapat berjalan menggunakan metode *inverse kinematics* dengan sensor IMU sebagai indikator kemiringan.

1.2 Rumusan Masalah

1. Bagaimana mekanisme kerja robot *humanoid* berbasis *inverse kinematics* terintegrasi sensor IMU.
2. Bagaimana mengembangkan kendali PID untuk mempertahankan kestabilan robot *humanoid* ketika berjalan.
3. Bagaimana performa robot *humanoid* berjalan berbasis *inverse kinematics* terintegrasi sensor IMU.

1.3 Tujuan Penelitian

1. Mengembangkan mekanisme kerja robot *humanoid* berbasis *inverse kinematics* terintegrasi sensor IMU.

2. Mengintegrasikan kendali PID untuk mempertahankan kestabilan robot *humanoid* ketika berjalan.
3. Menguji performa robot *humanoid* berjalan berbasis *inverse kinematics* terintegrasi sensor IMU

1.4 Manfaat Penelitian

1. Memberikan solusi berupa teknologi yang dapat mengatur robot *humanoid* ketika berjalan.
2. Sebagai sumber referensi perkembangan teknologi robot *humanoid*.
3. Dapat dimanfaatkan bagi masyarakat luas sebagai robot edukasi maupun robot industri dalam penggunaan metode *inverse kinematics*.

1.5 Batasan Masalah

1. Permukaan lintasan datar dan rata.
2. Pengujian yang dilakukan hanya pada robot kondo tipe KHR-1.
3. *Joint* yang digunakan ketika berjalan hanya 12 *joint*, 2 pada tangan dan 10 pada kaki.

BAB II TINJAUAN PUSTAKA

2.1 *State of Art*

Praditya Handi Setiawan (2016), berfokus dalam analisis *inverse kinematics* yang dilakukan secara tersegmentasi dengan menggunakan pendekatan geometris hingga mampu memberikan respon posisi setiap *end effector*. Konsep ini memberikan perspektif penting terkait pendekatan geometris hingga mampu memberikan respon posisi.

Chairunnas (2019), membahas tentang sistem kontrol robot penyeimbang berbasis Arduino menggunakan metode PID dengan komunikasi *bluetooth HC-05*. *software* yang di gunakan adalah Arduino IDE. Komponen yang di gunakan pada sistem ini yaitu Arduino Uno, MPU6050, *driver motor*, *bluetooth HC-05*. Sistem ini berjalan berdasarkan inputan dari sensor MPU6050 yang digunakan untuk mendapatkan nilai tengah pada sebuah keseimbangan robot. Jika nilai keseimbangan tidak sama dengan nilai tengah maka motor robot akan bergerak dan berusaha untuk menyeimbangkan robot. Pada sistem ini di kontrol menggunakan *bluetooth HC-05* sebagai pengontrol nilai tengah dan menetapkan nilai tengah.

Kurniawan, Triwiyatno, & Sumardi (2017), melakukan Penelitian dibidang robotika khususnya dalam robot *humanoid*. Tugas Akhir ini merancang sebuah robot *humanoid* yang memiliki 12 derajat kebebasan (DoF) dengan sensor MPU-6050 sebagai indikator kemiringan. Robot *humanoid* bergerak dengan menggunakan metode *inverse kinematics* dengan mengikuti koordinat yang diberikan. Metode *inverse kinematics* membantu menemukan nilai sudut yang menggerakkan servo melalui koordinat akhir.

Utomo, Setyaningsih & Iqbal (2020), meneliti sistem kendali lengan robot yang dapat mengendalikan lengan robot hanya dengan menggerakkan tangan dan menekan tombol. Penelitian ini menggunakan sensor MPU-6050 sebagai sensor sudut dan potensiometer sebagai pengatur gerakan robot, selain itu juga menggunakan Arduino Uno sebagai pengolah masukan sensor agar dapat mengendalikan lengan robot. Adapun robot yang digunakan adalah robot lengan 4 DoF (*Degree of Freedom*) dengan penggerak motor servo. Penelitian ini

menggunakan metode *research and development* yaitu dengan mempelajari dan mengembangkan penelitian sebelumnya.

Muh Anshar (2020), memperkenalkan teknik baru dalam implementasi belajar berjalan untuk robot berkaki dimana metode usulan menggunakan pengembangan Algoritma Evolusi, yaitu metode belajar terarah berbasis algoritma evolusi atau *directed evolutionary algorithm learning* yang disingkat DEAL. Pada penelitian ini menggunakan robot berkaki empat sebagai robot uji dengan menerapkan metode DEAL untuk gerak robot berkaki.

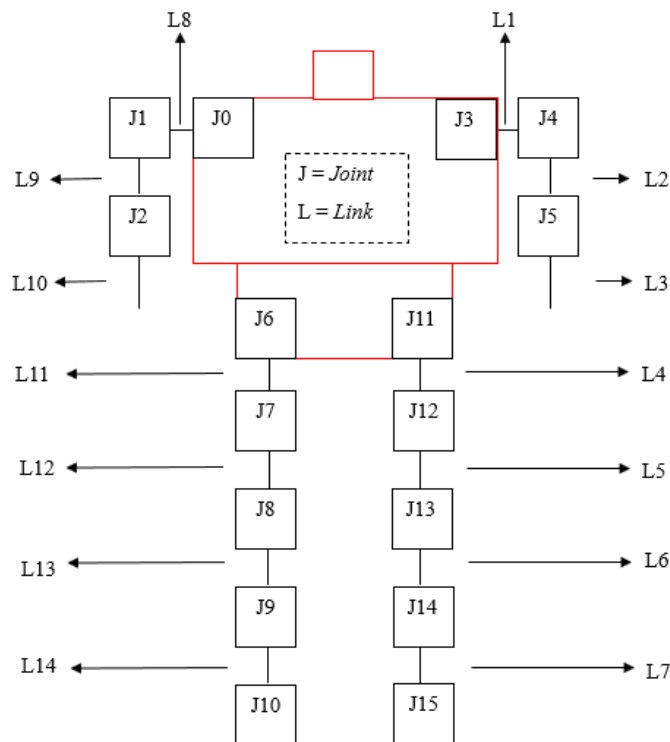
2.2 Robot *Humanoid*

Kata robot berasal dari bahasa Czech (Ceko), *robota* yang berarti pekerja atau kuli. Robot merupakan suatu perangkat mekanik yang mampu menjalankan pekerjaan fisik atas kendali manusia atau otomatis dengan program yang telah terpasang ataupun dengan kecerdasan buatan (*artificial intelligence*). Robot dapat diklasifikasikan berdasarkan fungsi fisiknya yaitu *non-mobile robot* dan *mobile robot*. *Non-mobile robot* merupakan robot yang tidak dapat berpindah secara keseluruhan dari posisinya, seperti robot lengan salah satu contoh *non-mobile robot*. Sedangkan *Mobile robot* merupakan robot yang dapat bergerak dengan berpindah secara keseluruhan bagian robot karena memiliki alat untuk bergerak. *Mobile robot* dapat diklasifikasi berdasarkan sistem penggerakannya yaitu robot beroda dan robot berkaki (Kurniawan & Sumardi, 2017). Robot beroda bergerak menggunakan roda yang digerakkan oleh aktuator, sedangkan robot berkaki menggunakan mekanik yang dibentuk sedemikian rupa layaknya mekanisme gerak seperti kaki makhluk hidup yang juga digerakkan oleh aktuator.

Untuk robot berkaki memiliki banyak variasi dalam jumlah kaki yang digunakan, seperti robot berkaki empat (*quadruped robot*) dan robot berkaki dua (*bipedal robot*). Dalam segi pengontrolan, robot berkaki 4 lebih mudah dilakukan karena dalam menjaga stabilitas selalu ada tiga atau dua kaki yang bisa menopang tubuhnya secara bersamaan, terutama saat bergerak (Muh. Anshar & Mary-Anne Williams, 2007). Hal ini memberikan keseimbangan yang lebih baik dibanding dengan robot berkaki dua yang jauh lebih sulit untuk menjaga stabilitas karena

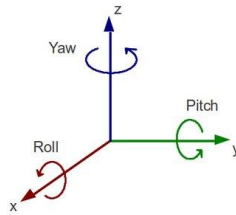
hanya memiliki dua titik kontak dengan tanah. Ini mengharuskan robot untuk secara aktif menyeimbangkan tubuhnya. Ketika dalam kondisi hanya satu kaki yang menopang saat bergerak, sehingga robot berkaki empat lebih stabil dan lebih mudah dikendalikan, namun kurang fleksibel dibandingkan dengan robot berkaki dua yang meskipun lebih sulit untuk dirancang dan dikendalikan, memiliki potensi mobilitas yang lebih tinggi dan lebih mirip dengan manusia. Sehingga dapat bergerak dengan cepat dan stabil.

Selanjutnya mengenai robot *humanoid* yang memiliki definisi dari *humanoid* yaitu segala sesuatu yang memiliki struktur menyerupai manusia. Maka robot *humanoid* dapat didefinisikan sebagai sebuah robot yang memiliki karakteristik menyerupai manusia, baik keseluruhan struktur maupun pergerakan dari robot tersebut. Riset mengenai analisis kinematika robot *humanoid* sudah mulai berkembang salah satunya ialah analisis *inverse kinematics* pada robot *humanoid* saat melakukan pergerakan tertentu (Praditya Handi Setiawan, 2016). Fokus pada tugas akhir ini adalah menyelesaikan analisis *inverse kinematics*. Dalam hal ini solusi *inverse kinematics* yang akan dibahas adalah menentukan sudut yang diinginkan pada masing-masing *joint* dari robot *humanoid*. Secara visual, struktur bentuk fisik pada robot *humanoid* tipe KHR-1 dapat dilihat pada Gambar 1.



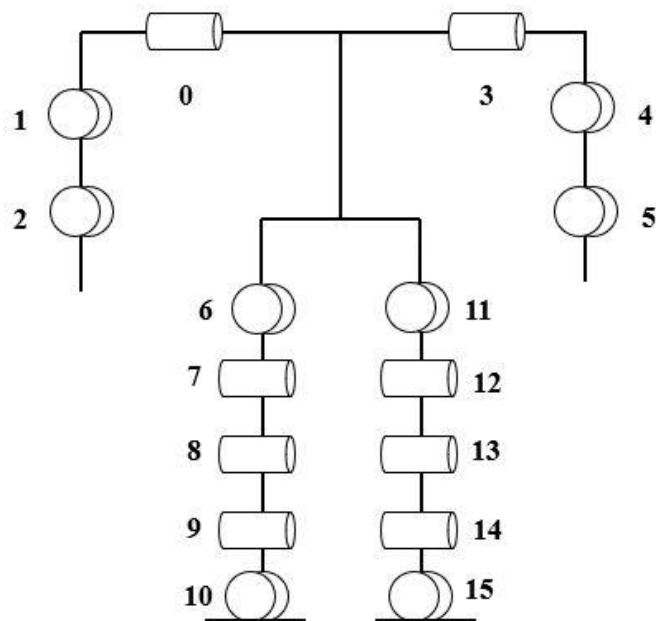
Gambar 1. Bagian – bagian robot *humanoid* jenis *bipedal*

Pada Gambar 1 dapat dijelaskan bahwa robot *humanoid* tipe KHR-1 memiliki 16 *joint* dan 14 *link*. Jumlah *joint* pada robot *humanoid* dapat dibagi menjadi 4 bagian yaitu : pada tangan terdapat 6 *joint*, dengan tangan kanan sebanyak 3 *joint* dan tangan kiri sebanyak 3 *joint*. Sedangkan, pada kaki terdapat 10 *joint*, kaki kanan sebanyak 5 *joint* dan kaki kiri sebanyak 5 *joint*. Terkhusus pada *joint* 9 dan *joint* 14 difokuskan sebagai penyeimbang berdasarkan *feedback* dari sensor IMU. Ketika melakukan gerakan berjalan, *joint* yang digunakan hanya 12 *joint* saja. *Joint* 1, 2, 4, 5 diabaikan sehingga *joint* 1 dan 2 menjadi sebuah *link* begitupun dengan *joint* 4 dan 5. Adapun acuan perputaran *joint* pada robot *humanoid* dapat dibagi kedalam 3 sumbu, yaitu : *roll* (x), *pitch* (y), dan *yaw* (z) yang dapat dilihat pada Gambar 2.



Gambar 2. Sumbu *roll*, *pitch* dan *yaw*

Pergerakan sumbu putar tiap *joint* pada robot *bipedal* dapat dilihat pada Gambar 3.



Gambar 3. Konfigurasi robot *humanoid*

Keterangan :

- a. *Joint* 0, 3 (*shoulder*) sumbu putarnya *pitch*.
- b. *Joint* 1,4 (*shoulder*) sumbu putarnya *roll*.
- c. *Joint* 2,5 (*elbow*) sumbu putarnya *roll*.
- d. *Joint* 6,11 (*hip*) sumbu putarnya *roll*.
- e. *Joint* 7,12 (*hip*) sumbu putarnya *pitch*.
- f. *Joint* 8,13 (*knee*) sumbu putarnya *pitch*.

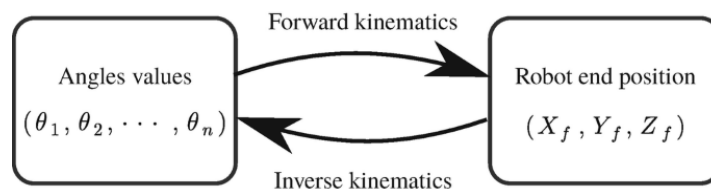
g. *Joint 9,14 (ankle)* sumbu putarnya *pitch*.

h. *Joint 10, 15 (ankle)* sumbu putarnya *roll*.

Sendi atau *joint* dari robot *humanoid* menggunakan jenis *revolute*. Sendi *revolute*, seperti sebuah engsel dan memungkinkan gerak rotasi antara dua *link*. Selain itu, robot *humanoid* ini memiliki kontroler Arduino dengan CPU tipe ARM Cortex. Kontroler tersebut terletak pada bagian punggung robot.

2.3 Kinematika Robot

Studi kinematika merupakan studi analitis pergerakan kaki atau lengan robot terhadap sistem kerangka koordinat acuan yang diam atau bergerak tanpa memperhatikan gaya yang menyebabkan pergerakan tersebut. Model kinematika merepresentasikan hubungan *end effector* dalam ruang tiga dimensi dengan variabel sendi. Analisis kinematika dibagi menjadi dua yaitu *forward kinematics* dan *inverse kinematics* (Widia & Risfendra, 2021). Gambar 4 merupakan konsep dari *forward kinematics* dan *inverse kinematics*.

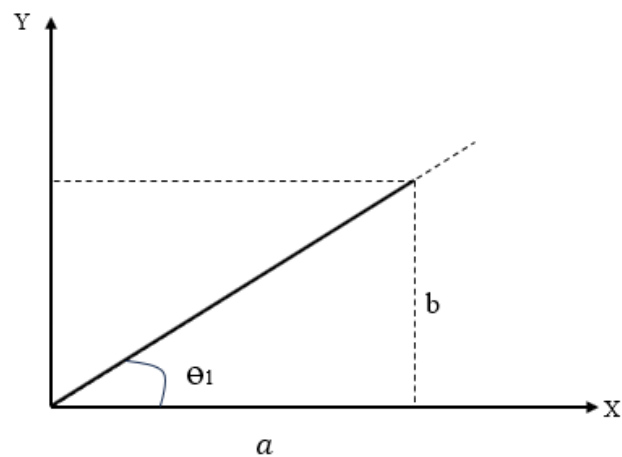


Gambar 4. Representasi *forward kinematics* dan *inverse kinematics*

Konsep *forward kinematics*, menggunakan input sudut dari setiap *joint* untuk mencari posisi dan orientasi *end-effector* sedangkan pada *inverse kinematics* menggunakan input posisi dan orientasi untuk mendapatkan nilai sudut dari setiap *joint*. Permasalahan yang dihadapi pada *inverse kinematics* adalah menentukan sudut setiap *joint* robot dengan posisi dan orientasi *end-effector* yang diketahui. Permasalahan *inverse kinematics* tidak seperti *forward kinematics* yang lebih sederhana, karena persamaannya yang *nonlinier*, dan juga adanya beberapa solusi yang muncul. Oleh karena itu, solusi dari analisis *inverse kinematics* disebut juga dengan *multiple solution*. Eksistensi dari setiap solusi dapat digunakan jika posisi *end-effector* dan orientasi yang diberikan termasuk dalam ruang kerja (*workspace*). Apabila posisi dan orientasi (titik tujuan) dari *end effector* berada di dalam area

jangkauan (*workspace*), maka sekurang-kurangnya terdapat satu solusi. Apabila solusinya ada, maka kemungkinan lain ialah solusinya lebih dari satu (*multiple solution*).

Solusi *inverse kinematics* berbasis geometris diperoleh berdasarkan sistem koordinat *link* dan pola geometris konfigurasi *joint* robot. Dalam penyelesaiannya, dapat menggunakan trigonometri untuk mendapatkan sudut-sudut dari *joint* robot (Andianto & Setyawan, 2018). Contoh penerapannya dapat dilihat pada analisis *inverse kinematics* dengan 1 *link* robot Planar pada Gambar 5.



Gambar 5. Manipulator planar dengan 1 *link*

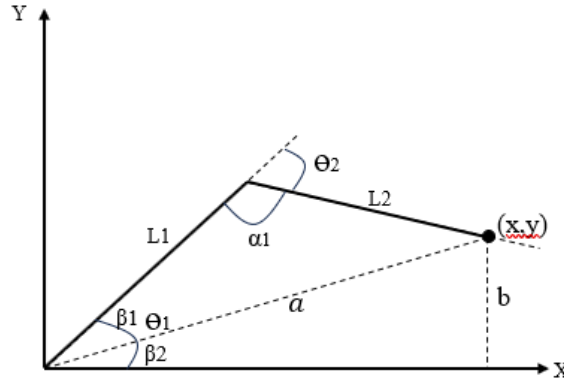
Dari Gambar 5, penyelesaian *inverse kinematics* dengan menggunakan pendekatan geometris yang terdapat pada Persamaan di bawah.

$$a = X, b = Y \quad (1)$$

$$\tan \theta_1 = \frac{b}{a} \quad (2)$$

$$\theta_1 = \tan^{-1}\left(\frac{b}{a}\right) \quad (3)$$

Selain itu, akan dijelaskan tentang konsep penyelesaian *inverse kinematics* menggunakan pendekatan geometris pada robot planar 2 *link*.



Gambar 6. Manipulator planar dengan 2 link

Untuk mengetahui nilai dari sudut dari *joint* pada robot planar 2 link, perhitungan dimulai dengan menghitung nilai Θ_2 . Metode geometris untuk mendapatkan nilai Θ_2 akan dituliskan dalam Persamaan berikut.

$$a = \sqrt{b^2 + c^2} \quad (4)$$

$$a^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos \alpha_1 \quad (5)$$

$$2L_1L_2 \cos \alpha_1 = L_1^2 + L_2^2 - a^2 \quad (6)$$

$$\cos \alpha_1 = \frac{L_1^2 + L_2^2 - a^2}{2L_1L_2} \quad (7)$$

$$\alpha_1 = \cos^{-1}\left(\frac{L_1^2 + L_2^2 - a^2}{2L_1L_2}\right) \quad (8)$$

$$\theta_2 = 180 - \alpha_1 \quad (9)$$

Setelah memperoleh nilai dari Θ_2 maka, nilai dari Θ_1 dapat diperoleh seperti pada Persamaan berikut.

$$L_2^2 = a^2 + L_1^2 - 2L_1a \cos \beta_1 \quad (10)$$

$$2L_1a \cos \beta_1 = a^2 + L_1^2 - L_2^2 \quad (11)$$

$$\cos \beta_1 = \frac{a^2 + L_1^2 - L_2^2}{2L_1a} \quad (12)$$

$$\beta_1 = \cos^{-1}\left(\frac{a^2 + L_1^2 - L_2^2}{2L_1a}\right) \quad (13)$$

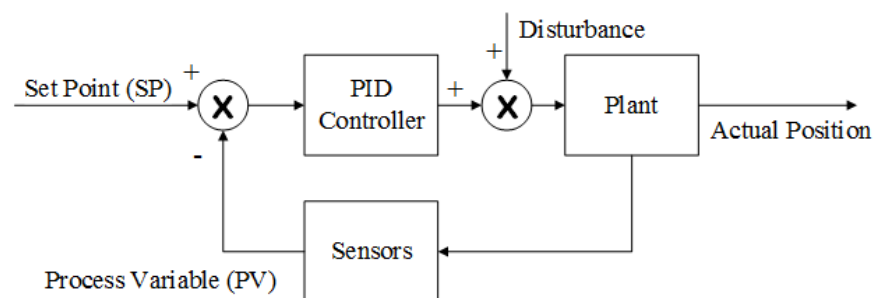
$$\sin \beta_2 = \frac{b}{a} \quad (14)$$

$$\beta_2 = \sin^{-1}\left(\frac{b}{a}\right) \quad (15)$$

$$\theta_1 = \beta_1 - \beta_2 \quad (16)$$

2.4 Kendali PID

Kendali PID adalah pengendali yang sering digunakan dibidang industri, robotika maupun *home automation*. Kendali PID merupakan suatu kendali yang sederhana dan dikategorikan sebagai kendali *Single Input single Output* (SISO). Kendali merupakan kendali daur tertutup dan bekerja dengan cara mengoreksi keluaran suatu sistem dengan melakukan perhitungan selisih antara *setpoint* atau sinyal masukan dan keadaan sistem saat ini yang diumpan balik sehingga memperoleh nilai *error* kemudian nilai *error* dikalkulasi pada pengendali PID, sehingga mengeluarkan sinyal berupa usaha untuk mengurangi nilai *error* pada sistem tersebut (Riyanto dkk., 2021). Diagram kendali PID dapat dilihat pada Gambar 7.



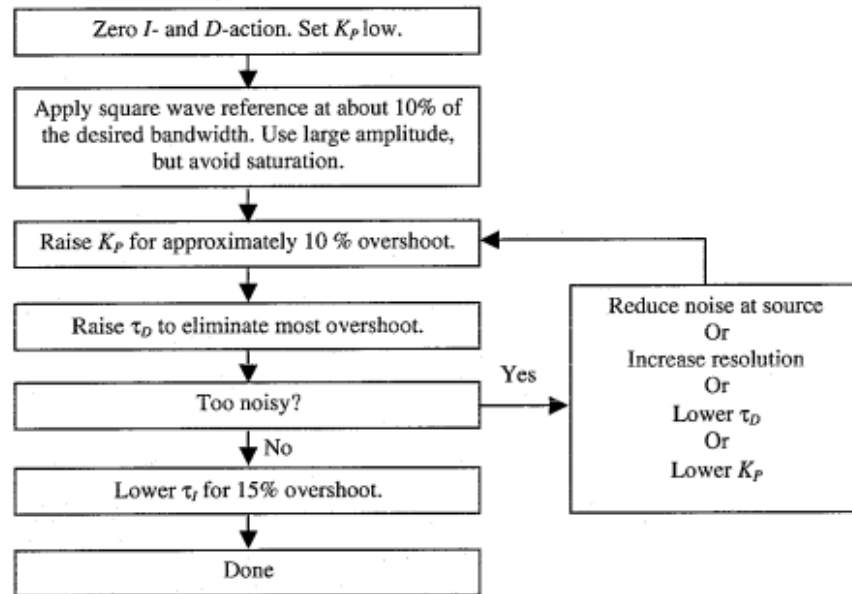
Gambar 7. Diagram sistem kendali PID

PID bekerja dengan tiga komponen pengendali utamanya yaitu komponen *proportional*, *integral*, dan *derivative*. *Proportional* memiliki keluaran berdasarkan nilai *error* saat ini. Komponen *integral* menghasilkan keluaran berdasarkan jumlah *error* saat ini dan *error* sebelumnya. Sedangkan komponen *derivative* memiliki keluaran berdasarkan selisih *error* saat ini dan *error* sebelumnya. Keluaran yang dihasilkan tiap komponen akan dijumlahkan sehingga menghasilkan bobot sinyal keluaran $u(t)$ yang akan di umpan pada *plant* atau sistem yang akan dikendalikan.

Untuk mendapatkan kendali PID yang memiliki performa baik dibutuhkan pengaturan gain atau konstanta yang tepat pada komponen *proportional* (K_p), *integral* (K_i), dan *derivative* (K_d) (Visioli, 2006). Dalam pengaturan konstanta PID ada beberapa hal yang perlu diperhatikan dari *respon* keluaran sistem yang dikendalikan, berikut beberapa parameter yang perlu diperhatikan.

1. *Rise time*, merupakan waktu yang dibutuhkan sistem untuk mencapai dari kondisi awal menuju nilai *setpoint*.
2. *Overshoot*, merupakan nilai maksimum dari *respon* sistem ketika menuju stabil pada *setpoint*.
3. *Settling time*, didefinisikan sebagai waktu yang dibutuhkan sistem hingga mencapai kondisi *steady state*.
4. *Steady-state error*, merupakan selisih antara keluaran sistem saat kondisi *steady state* dengan *setpoint*.
5. *Stability*, merupakan *respon* suatu sistem untuk tetap bertahan pada posisi *steady-state* ketika mencapai *setpoint*.
6. *Steady state*, kondisi sistem telah mencapai kestabilan.

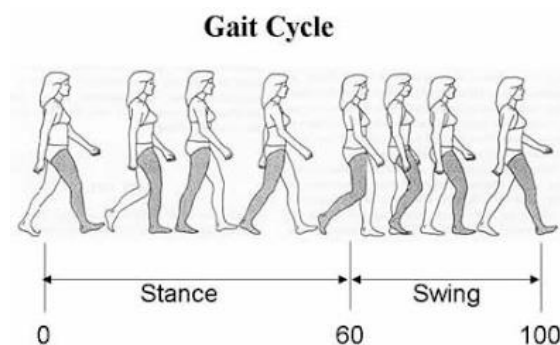
Untuk mendapatkan respon sistem yang baik diperlukan sebuah pengaturan atau *tuning* pada konstanta PID. Beberapa metode *tuning* banyak dijumpai yang telah sering diaplikasikan pada dunia industri. Salah satunya adalah metode *manual tuning* atau *trial and error method* yaitu dengan cara melakukan percobaan *tuning* berulang-ulang kali secara manual hingga diperoleh konstanta yang memiliki respon sistem terbaik. Adapun tahap-tahap manual *tuning* yang umum digunakan dapat dilihat pada Gambar 8.



Gambar 8. Metode Manual *Tuning* Konstanta PID

2.5 Perencanaan Gaya Berjalan

Robot *humanoid* pada umumnya memiliki dua kaki, sehingga dapat disebut dengan istilah *biped* robot yaitu robot yang memiliki dua kaki. Konsep gaya berjalan pada robot bipedal mengikuti pola berjalan manusia. Gaya normal berjalan manusia dibagi menjadi tahap berdiri (*stance*) diperkirakan (60%) dan tahap berjalan (*swing*) sebesar (40%). *Stance* adalah pada saat kaki berada di tanah seperti tampak pada Gambar 9. *Swing* adalah kondisi saat kaki melangkah untuk persiapan ke tahap *stance* selanjutnya. Jika panjang dan kecepatan melangkah dikurangi dapat memungkinkan untuk menambah stabilitas *bipedal* (Kurniawan & Sumardi, 2017).

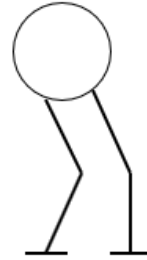


Gambar 9. Gaya berjalan manusia

Selain itu, pola gaya berjalan dapat dibagi menjadi 2 yaitu *single support phase* dan *double support phase*. *Single support phase* adalah saat satu kaki menjadi tumpuan titik berat ketika satu kaki yang lain melangkah kedepan, hal tersebut seperti yang terlihat pada Gambar 10 bagian (a). Sedangkan *double support phase* adalah kondisi ketika kaki yang sebelumnya melangkah (di *single support phase*) telah menapak ke tanah dan satu kaki yang lainnya juga menapak, hal tersebut seperti yang tertera pada Gambar 10 bagian (b). Untuk mendapatkan gaya berjalan yang dinamis maka perpindahan antara *single support phase* dan *double support phase* harus sangat halus.



(a)



(b)

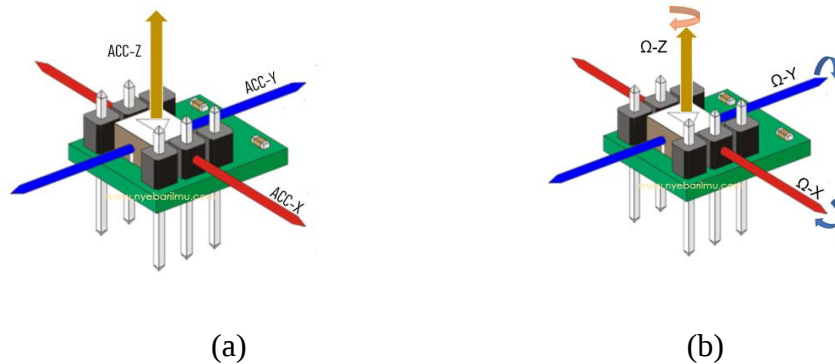
Gambar 10. Support Phase Pada Robot *Humanoid*

2.6 IMU (Inertial Measurement Unit) Sensor

Inertial Measurement Unit merupakan suatu unit pengukuran kecepatan, arah serta posisi suatu benda dengan menggunakan *gyroscope* dan *accelerometer*. Seperti tampak pada Gambar 11 *gyroscope* dan *accelerometer* masing-masing memiliki 3 sumbu derajat kebebasan. *Gyroscope* sendiri merupakan unit pengukur kecepatan sudut dan posisi sudut dalam tiga sumbu derajat kebebasan, sedangkan *accelerometer* ialah unit pengukur kecepatan/percepatan linier serta perubahan posisi dan arah dalam tiga sumbu derajat kebebasan (Kurniawan & Sumardi, 2017)..

Sensor IMU dijumpai pada navigasi pesawat dan mulai banyak digunakan pada berbagai bidang industri dan dalam kehidupan sehari-hari seperti, pada navigasi pesawat, navigasi roket, navigasi satelit, pada *mobile robot*, bahkan *smartphone* hingga *camera gimbal*.

IMU bekerja dengan menggunakan prinsip *micro-electro-mechanical system* (MEMS) di mana sebuah penghantar bekerja secara mekanis dan elektris, perubahan keadaan mekanis yang diakibatkan oleh gaya dari lingkungan mengakibatkan perubahan perilaku elektris sehingga menghasilkan pengukuran dalam sinyal listrik. Dapat diumpamakan pada dua buah batang bermuatan yang diletakkan pada sebuah bidang datar, ketika bidang tersebut dimiringkan maka salah satu batang akan bergerak mendekati batang lainnya sehingga terjadi perubahan perilaku elektris berupa nilai kapasitansi antara kedua batang tersebut berubah. Sistem MEMS ini dibuat dalam bentuk mikro yang terintegrasi dalam satu paket *chip*. Berikut adalah salah satu contoh struktur elektro-mekanis dari IMU (*Inertial measurement unit*) sensor.



Gambar 11. (a) Axis pada accelerometer (b) axis pada gyroscope
Salah satu jenis IMU sensor adalah MPU6050 produk dari *Invensense* yang memiliki *gyroscope* dan *accelerometer* masing-masing dalam tiga sumbu derajat kebebasan. Keluaran dari *gyroscope* dan *accelerometer* dapat dilakukan kalkulasi untuk mendapatkan posisi sudut absolut dalam tiga sumbu derajat kebebasan. Bentuk fisik sensor MPU6050 dapat dilihat pada Gambar 12.



Gambar 12. MPU6050

2.7 Filter

Filter atau penyaring merupakan sebuah sistem yang bertujuan untuk meloloskan sinyal dibutuhkan pada frekuensi tertentu dan menghambat sinyal pada rentang frekuensi tertentu. Filter dapat diterapkan dalam bentuk *hardware* maupun dapat diterapkan dalam bentuk *software* (Sani dkk., 2024). Filter sering diaplikasikan pada rangkaian pengkondisian sinyal keluaran sensor, rangkaian pemancar dan penerima radio frekuensi maupun pada rangkaian elektronika daya.

High-pass filter atau biasa disebut HPF adalah merupakan salah satu jenis filter atau penyaring yang melewatkan sinyal frekuensi tinggi dan menghambat sinyal frekuensi rendah. HPS sering digunakan untuk menapis sinyal yang mengalami gangguan frekuensi rendah seperti sinyal keluaran dari *gyroscope* yang memiliki *drift* atau perubahan sinyal secara perlahan.

Low-pass filter atau biasa disebut LPF adalah merupakan salah satu jenis filter atau penyaring yang melewatkan sinyal frekuensi rendah dan menghambat sinyal frekuensi tinggi. HPS sering digunakan untuk menapis sinyal yang mengalami gangguan frekuensi rendah contohnya seperti sinyal keluaran dari *accelerometer* yang sangat sensitif sehingga memiliki banyak derau frekuensi tinggi.

Complementary filter merupakan salah satu metode filter sinyal untuk menghilangkan *noise* atau derau yang terdapat pada sinyal. Salah satu penggunaan *Complementary filter* yang sering digunakan adalah untuk memfilter sinyal *gyroscope* dan *accelerometer* dari IMU sensor untuk mendapatkan sinyal *roll pitch yaw* yang terbebas dari derau. *Complementary filter* terdiri dari *low pass filter* yang akan meloloskan sinyal frekuensi rendah, sedangkan *blocking sinyal* frekuensi tinggi terdapat pada sinyal *accelerometer*, dan terdiri dari *high pass filter* yang akan meloloskan sinyal frekuensi tinggi sedangkan *blocking* sinyal frekuensi rendah dilakukan pada output sinyal *gyroscope*.