

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian - penelitian mengenai deteksi posisi pada robot 3 roda *omnidirectional* sudah pernah dilakukan, sehingga permasalahan yang ada dijadikan sebagai acuan, penambahan dan referensi dalam pengembangan sistem yang ingin dirancang pada alat ini. Berikut beberapa penelitian yang sudah dilakukan.

Tabel 2. 1 Tinjauan Pustaka

No	Judul	Masalah	Metode	Hasil
1.	“Penerapan kinematika untuk lokalisasi pada robot sepak bola beroda” ditulis oleh anan pepe abseno.[6]	Membuat lokalisasi robot yang bisa bergerak berdasarkan Koordinat.	Kinematika Odometry.	Rata-rata <i>error</i> yang diketahui pada saat robot melakukan pergerakan , diketahui oleh sistem sebesar 0,399 cm sedangkan rata-rata <i>error</i> sebenarnya sebesar 2,108 cm.
2.	"A Modified Kalman Filter-Based Mobile Robot Position	Masalah yang ingin diatasi adalah estimasi posisi robot bergerak di	Penerapan Filter Kalman. Filter Kalman digunakan untuk menggabungk	Metode yang diusulkan menghasilkan estimasi posisi robot dengan

	<i>Measurement using an Accelerometer and Wheels Encoder"</i> ditulis oleh Ali Madhloom, Firas Raheem, dan Azad Kareem.[7]	lingkungan tertutup menggunakan sensor encoder dan akselerometer.	an pembacaan sensor akselerometer dan odometri guna memperkirakan posisi dan kecepatan robot bergerak.	kesalahan yang diterima, yaitu kurang dari 0,2 meter, baik untuk jalur yang mudah maupun sulit.
3.	"Robot Keseimbangan dengan Kendali Proporsional-Integral-Derivative (PID) dan Kalman Filter" ditulis oleh Alfian Maarif, Ricky Dwi Puriyanto, Fadlur Rahman T Hasan.[8]	Robot Keseimbangan memiliki dinamika cepat, tidak stabil, dan sifat nonlinier. Sensor akselerometer yang digunakan sensitif dan bernoise.	Digunakan pengendali PID untuk mengendalikan perubahan sudut dan Kalman filter untuk mengurangi noise pada sensor akselerometer.	Kalman filter berhasil mengurangi noise pada sensor akselerometer. Parameter terbaik untuk Kalman filter adalah $R = 10$ dan $Q = 0.01$. Pengendali PID berhasil menstabilkan robot dengan parameter terbaik $K_p = 20$, $K_i = 1$, dan $K_d = 20$.
3.	<i>"Noise Reduction in the Accelerometer and</i>	Masalah utama yang dihadapi adalah kebisingan	Metode yang digunakan adalah filter Kalman untuk mengurangi	Filter Kalman berhasil mengurangi kebisingan

	<i>Gyroscope Sensor with the Kalman Filter Algorithm</i> ” oleh Rio Ikhsan Alfian, Alfian Ma’arif, Sunardi.[5]	pada keluaran sensor accelerometer dan gyroscope.	kebisingan. Nilai parameter Q dan R dalam algoritma Kalman divariasikan untuk menemukan nilai optimal.	pada sensor. Penelitian menunjukkan bahwa nilai $Q < R$ memberikan hasil terbaik dalam mengurangi kebisingan.
--	--	---	--	---

Berdasarkan tinjauan pustaka Tabel 2.1 tersebut metode Kalman Filter sudah pernah digunakan akan tetapi metode kalman filter digunakan untuk memfilter data *accelerometer* seperti pada[5][8][7]. Pada tugas akhir ini permasalahan ada pada penentuan arah hadap robot yang digunakan untuk *setpoint* meluruskan arah pada saat *navigasi*. Untuk itu pada tugas akhir ini merancang metode Kalman Filter untuk memproses data mentah z axis dari sensor imu yang nantinya diterapkan dalam *prototype* robot KRSBI-B untuk membantu pergerakan pada arena yang sudah ditentukan dalam koordinat kartesius, selain penggunaan metode Kalman Filter tugas akhir ini juga merancang *basestation* untuk memonitoring data keluaran sensor, yang diharapkan dapat mempermudah proses analisis dan penanganan masalah pada robot.

2.2 Landasan Teori

2.2.1 Deteksi Posisi

Deteksi posisi pada robot beroda merujuk pada kemampuan sistem untuk menentukan atau mengidentifikasi letak atau posisi robot secara akurat. Hal ini dapat dilakukan dengan menggunakan berbagai teknik dan sensor yang terintegrasi dalam robot. Deteksi posisi penting dalam navigasi dan pengendalian robot, karena dengan mengetahui posisinya, robot dapat melakukan tugas-tugasnya dengan efektif dan efisien[3]. Beberapa metode yang umum digunakan dalam deteksi posisi pada robot beroda meliputi penggunaan sensor pengukuran jarak, sensor inersial seperti akselerometer dan giroskop, pemrosesan citra untuk analisis lingkungan, dan teknik pemetaan simultan dan pemetaan lokal

(SLAM). Dengan deteksi posisi yang akurat, robot beroda dapat memperoleh informasi tentang lokasinya dalam lingkungan sekitar, memperbaiki navigasi, dan mengambil keputusan yang tepat saat bergerak atau berinteraksi dengan objek di sekitarnya.

Dalam deteksi posisi terdapat beberapa parameter yang digunakan untuk menghitung Koordinat posisi robot. Seperti menggunakan rotary encoder yang dikombinasikan dengan roda untuk menghitung jarak tempuh robot. Perhitungan jarak tempuh robot bisa digunakan persamaan berikut, keliling roda free wheel (KW) dapat dihitung dengan mengalikan diameter roda *freewheel* (DW) dengan nilai pi (π), lihat persamaan (1). Perubahan Koordinat posisi x (X_{tempuh}) dihitung dengan membagi jumlah putaran pulsa (*pulse*) dengan resolusi encoder ($resolusi_{enc}$) dan dikalikan dengan keliling roda *freewheel* pada sumbu x (KW_x), persamaan (2). Sedangkan perubahan Koordinat posisi y (Y_{tempuh}) dihitung dengan rumus yang sama, hanya menggantikan KW_y dengan keliling roda *freewheel* pada sumbu y (KW_y), persamaan (3)[9]. Jadi, persamaan keseluruhan adalah sebagai berikut:

$$KW = DW \times \pi \quad (1)$$

$$X_{tempuh} = \left(\frac{pulse(x)}{resolusi_{enc}} \right) \times KW_x \quad (2)$$

$$Y_{tempuh} = \left(\frac{pulse(y)}{resolusi_{enc}} \right) \times KW_y \quad (3)$$

2.2.2 Robot holonomic

Robot holonomic adalah jenis robot yang memiliki kemampuan untuk melakukan gerakan translasi dan rotasi secara independen, sehingga memungkinkan robot untuk bergerak dengan lebih bebas dan fleksibel. Dalam hal ini, robot holonomic dapat melakukan gerakan translasi pada semua arah dan rotasi di tempat tanpa harus bergerak maju atau mundur[2]. Sementara itu, robot nonholonomic hanya dapat melakukan gerakan translasi ke depan dan belakang serta rotasi, namun tidak dapat melakukan gerakan translasi ke samping.

Robot *holonomic* biasanya digunakan pada aplikasi yang memerlukan *maneuver* yang kompleks dan bebas, seperti robot pengangkut barang, mobil otonom, dan sejenisnya. Beberapa contoh jenis

robot *holonomic* yang umum diantaranya robot roda *omnidirectional* dan robot roda mecanum. Robot holonomic 3 roda omnidirectional adalah salah satu jenis robot holonomic yang memiliki tiga roda yang ditempatkan pada suatu sumbu yang sejajar dengan arah gerakan robot. Ketiga roda tersebut dapat berputar secara independen dan memungkinkan robot untuk melakukan gerakan translasi dan rotasi secara bebas dan fleksibel. Keuntungan dari robot holonomic 3 roda adalah kemampuan untuk bergerak pada ruang yang sempit dan melakukan *manuever* yang rumit dengan cepat.

2.2.3 Kalman filter

Kalman Filter adalah sebuah metode matematika yang digunakan untuk mengestimasi kondisi sistem yang tidak terlihat secara langsung dengan menggabungkan informasi dari beberapa sensor dan model sistem. Kalman filter digunakan untuk memperkirakan keadaan robot atau objek yang diamati dari data sensor, seperti posisi, kecepatan, dan percepatan[2]. Metode ini dikembangkan oleh Rudolf Kalman pada tahun 1960-an dan digunakan secara luas dalam berbagai aplikasi, seperti navigasi pesawat terbang, kendali robot, dan pengolahan sinyal.

Kalman Filter bekerja dengan memprediksi kondisi sistem berdasarkan model sistem dan informasi yang diberikan oleh sensor. Kemudian, Kalman Filter memperbaiki prediksi tersebut dengan menggabungkan informasi dari sensor dan model sistem yang digunakan. Dalam pengolahan data sensor, Kalman Filter menggunakan dua tahap utama yaitu tahap prediksi lihat persamaan (4) dan tahap koreksi lihat persamaan (5). Ada beberapa metode kalman filter, tetapi pada tugas akhir ini menggunakan Kalman Filter standar, metode tersebut sudah cukup untuk mereduksi atau mengurangi *noise* pada sensor[10]. Persamaan kalman filter ditunjukkan dibawah, tetapi dalam prosesnya persamaan tersebut bisa dimodifikasi sesuai kebutuhan.

Prediksi: (4)

$$\hat{x}_k^- = A * \hat{x}_{k-1} + B * u_{k-1}$$

$$P_k^- = A * P_{k-1} * A' + Q$$

Koreksi: (5)

$$S = H * P_k^- * H' + R$$

$$K_k = P_k^- * H' * S^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k * (z_k - H * \hat{x}_k^-)$$

$$P_k = (I - K_k * H) * P_k^-$$

Dimana \hat{x}_k^- merupakan estimasi awal keadaan sistem di waktu ke- k sebelum pengamatan baru diperhitungkan, di mana B dan u_{k-1} menunjukkan bagaimana pengendalian mempengaruhi estimasi tersebut. Pada prediksi, Q mencerminkan ketidakpastian perubahan keadaan seiring waktu. Saat update S adalah inovasi kovarian, yang mengukur sejauh mana pengukuran aktual z_k cocok dengan prediksi $H * \hat{x}_k^-$ berdasarkan ketidakpastian yang dihitung dari tahap prediksi. H adalah matriks pengamatan yang menghubungkan variabel yang diamati dengan perkiraan. K_k adalah matriks Kalman Gain, yang menunjukkan sejauh mana kita harus menggabungkan informasi dari pengukuran aktual dengan prediksi kita. Ini menggambarkan bobot antara pengukuran aktual dan prediksi. S^{-1} adalah invers dari matriks S . Kemudian \hat{x}_k Ini adalah perkiraan akhir yang diperbarui setelah menggabungkan informasi pengukuran aktual z_k dengan prediksi $H * \hat{x}_k^-$ berdasarkan matriks Kalman Gain. Ini adalah hasil akhir dari langkah koreksi. Terakhir adalah P_k Ini adalah estimasi kovarian yang diperbarui setelah langkah koreksi. Ini menggambarkan ketidakpastian setelah menggabungkan informasi pengukuran aktual[11].

2.2.4 PID

Kontrol PID adalah kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri untuk mengatur nilai kesalahan antara setpoint dan variabel proses terukur. Kontrol PID merupakan gabungan antara kontrol proporsional, kontrol integral, dan kontrol derivatif. Kontrol PID memberikan stabilitas, respon yang cepat, dan *error* steady state yang kecil atau bahkan hilang[12]. Rumus perhitungan kontrol PID bisa dilihat pada persamaan (10).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (6)$$

Di mana :

- $u(t)$ adalah adalah sinyal kontrol yang dihasilkan oleh kontroler PID pada waktu t .
- K_p, K_i, K_d adalah konstanta pengaturan yang disebut sebagai gain proporsional, integral, dan derivatif, masing-masing.
- $e(t)$ adalah *error* pada waktu t , yang merupakan selisih antara setpoint (nilai yang diinginkan) dan nilai aktual variabel yang diatur.
- $\int_0^t e(\tau) d\tau$ adalah integral dari *error* dari awal waktu hingga waktu t , yang membantu mengatasi kesalahan sistem steady-state.
- $\frac{de(t)}{dt}$ adalah turunan *error* terhadap waktu, yang membantu merespon perubahan cepat dalam *error*.

2.2.5 RMSE (Root Mean Square Error)

Untuk mempermudah analisis hasil, diperlukan suatu pendekatan yang dapat mengukur kesalahan total atau “total *error*” antara data acuan dan data observasi. Dalam hal ini, untuk mengukur total kesalahan yang terkait dengan pengujian tugas akhir ini, digunakan metode RMSE (Root Mean Square *Error*). Total *error* yang dihitung oleh RMSE adalah dalam bentuk rata-rata kesalahan kuadrat. Pemilihan metode RMSE untuk mengukur total kesalahan didasarkan pada alasan bahwa ada beberapa data dengan nilai acuan 0.

RMSE adalah metrik evaluasi dalam analisis statistika dan prediksi yang mengukur perbedaan antara prediksi dan observasi sebenarnya. Ini dihitung dengan mengambil akar kuadrat dari rata-rata selisih kuadrat antara prediksi dan observasi. RMSE membantu menilai akurasi prediksi atau model, dengan nilai yang lebih rendah menunjukkan prediksi yang lebih akurat terhadap data sebenarnya[13]. Rumus perhitungan dari RMSE sebagaimana dinyatakan dalam persamaan (7).

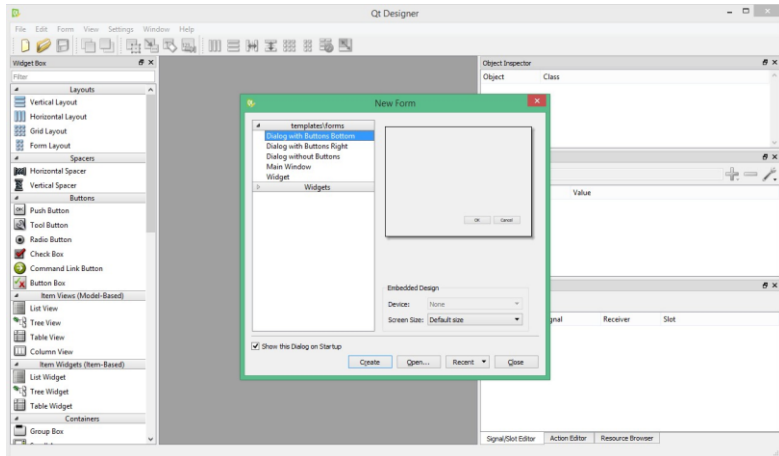
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

Dimana :

- n adalah jumlah data atau titik pengamatan.
- i adalah indeks yang mengacu pada setiap data atau titik pengamatan
- y_i adalah nilai observasi sebenarnya untuk sampel data $ke - i$
- \hat{y}_i adalah nilai data acuan untuk data $ke - i$

2.2.6 PyQt5

PyQt5 adalah sebuah framework yang sering digunakan dalam bahasa pemrograman Python untuk mengembangkan antarmuka pengguna yang intuitif. Dalam tugas akhir ini, penulis memanfaatkan PyQt5 untuk menciptakan sebuah *basestation* yang memungkinkan pengguna untuk berinteraksi dengan robot melalui perintah yang mudah dan juga memberikan kemampuan untuk melihat status robot secara *real-time*. *Basestation* ini menjadi alat yang sangat berguna dalam mengendalikan robot dan memantau keadaannya. Dengan menggunakan fitur-fitur PyQt5, penulis dapat membuat tampilan antarmuka yang menarik dan responsif, sehingga pengguna dapat dengan mudah berkomunikasi dengan robot dan memperoleh informasi yang dibutuhkan secara *real-time*[14]. Gambar 2.1 merupakan antarmuka awal ketika membuka software Qt designer.



Gambar 2. 1 tampilan awal pyqt5

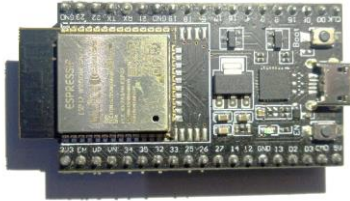
2.2.7 Micropython

MicroPython adalah versi Python yang dioptimalkan untuk perangkat kecil seperti Raspberry Pico. Saya menggunakannya untuk memprogram robot saya. Dengan MicroPython, saya bisa menggunakan bahasa pemrograman Python yang mudah dipahami untuk mengontrol pergerakan robot, membaca sensor, dan berkomunikasi dengan perangkat lain. MicroPython dirancang khusus agar bisa berjalan di perangkat dengan sumber daya terbatas. Saya menggunakan bahasa pemrograman ini untuk memprogram robot saya, micropython menjadi pilihan karena kesederhanaannya dalam membuat kode programnya[15].

2.2.8 ESP32

ESP32 adalah mikrokontroler yang sangat populer dan serbaguna, dikembangkan oleh perusahaan Tiongkok, Espressif Systems. Keunggulan utamanya terletak pada kemampuan nirkabelnya, dengan dukungan Wifi dan Bluetooth, memungkinkan perangkat terhubung ke internet dan berkomunikasi secara nirkabel dengan perangkat lain. Selain itu, ESP32 juga memiliki dua inti prosesor (dual-core) Xtensa LX6 yang kuat dengan kecepatan hingga 240 MHz. Selain itu, ESP32 memiliki banyak GPIO yang dapat difungsikan sesuai kebutuhan kita[16]. Hal ini membuat ESP32 menjadi pilihan untuk menjadikannya kontroler utama

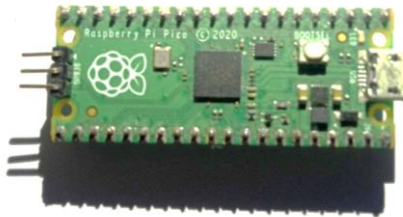
dalam pembuatan tugas akhir ini, bentuk ESP32 bisa dilihat pada Gambar 2.2.



Gambar 2. 2 ESP32

2.2.9 Raspberry Pico

Raspberry Pico merupakan development board yang dikembangkan oleh Raspberry Foundation. Pi Pico menggunakan RP2040 sebagai mikrokontrolernya yang memiliki arsitektur Arm Cortex M0+ dengan kecepatan clock hingga 133 Mhz[17]. Pada Tugas Akhir ini penulis menggunakan raspberry pico sebagai pembaca sensor sekaligus digunakan untuk menjalankan metode kalman filternya. Bentuk dari Raspberry Pico bisa dilihat pada Gambar 2.3.



Gambar 2. 3 Raspberry pico

2.2.10 Rotary Encoder

Rotary encoder terdiri dari sebuah piringan tipis dengan lubang-lubang melingkar di dalamnya. Di satu sisi piringan, terdapat LED yang memancarkan cahaya ke arah piringan, sedangkan di sisi lainnya terdapat photo-transistor yang mampu mendeteksi cahaya yang menjebak piringan tersebut dari LED yang berada di sisi berlawanan. Piringan tipis ini

terhubung dengan poros motor atau mekanisme berputar lainnya yang ingin kita pantau posisinya. Ketika motor berputar, piringan juga ikut berputar.

Ketika posisi piringan menyebabkan cahaya dari LED dapat mencapai phototransistor melalui lubang-lubang yang ada, maka phototransistor akan jenuh dan menghasilkan pulsa gelombang persegi. Pada tugas akhir ini menggunakan rotary encoder dengan spesifikasi 600 ppr bentuk dari sensor rotary encoder yang digunakan bisa dilihat pada Gambar 2.4.



Gambar 2. 4 Rotary Encoder

Rotary encoder pada penulisan tugas akhir ini digunakan untuk menghitung jarak dalam satuan cm dengan rumus konversi *pulse* ke *centimeter* (cm) sebagai berikut:

$$KelilingRoda = \pi \times d \quad (8)$$

$$Centimeter = \frac{pulse}{(ppr \div KelilingRoda)} \quad (9)$$

Keterangan :

- π : 3,14
- d : Diameterlingkarang(Roda)
- *Pulse* : Pulsa sinyal sensor rotary encoder
- *ppr* : PulsePer – rotation

2.2.11 IMU (*Inertial Measurement Unit*) GY-87

Sensor IMU GY-87 adalah sebuah sensor *inertial measurement unit* (IMU) yang terdiri dari tiga sensor yaitu *accelerometer*, *gyroscope*, dan *magnetometer* dalam satu modul. Modul sensor ini dapat digunakan untuk mengukur percepatan, kecepatan sudut, dan orientasi suatu benda dalam tiga dimensi. *Accelerometer* pada sensor ini dapat mengukur percepatan pada tiga sumbu, yaitu sumbu X, Y, dan Z. Sedangkan *gyroscope* digunakan untuk mengukur kecepatan sudut pada tiga sumbu yang sama. Dengan kedua sensor tersebut, sensor IMU GY-87 dapat memberikan informasi mengenai gerakan objek dalam tiga dimensi[18]. Bentuk dari sensor IMU GY-87 bisa dilihat pada Gambar 2.5.



Gambar 2. 5 Modul Sensor IMU GY-87

2.2.12 Motor Driver BTS7960

Motor driver BTS7960 adalah modul elektronik yang digunakan untuk mengontrol motor DC dengan daya tinggi, modul ini dapat mengontrol motor DC dengan arus hingga 43A dan tegangan hingga 27 V. Modul BTS7960 dilengkapi dengan beberapa fitur seperti perlindungan over-current, over-heat, dan under-voltage[19]. Bentuk dari Motor Driver BTS7960 bisa dilihat pada Gambar 2.6.



Gambar 2. 6 Motor Driver BTS7960

2.2.13 LCD I2C 16x2

LCD (Liquid Crystal Display) digunakan secara umum untuk menampilkan teks, angka, atau simbol tertentu. LCD ini memiliki kemampuan untuk menampilkan karakter dengan ukuran 16 x 2. Dengan menggunakan penampil LCD 16x2, pengguna dapat memantau sensor atau melacak jalannya program. LCD 16x2 bisa dilihat pada Gambar 2.8.



Gambar 2. 7 LCD 16 x2

2.2.14 Motor DC

Jenis motor listrik yang mengubah energi listrik menjadi energi mekanik melalui prinsip elektromagnetik, dengan energi listrik yang digunakan adalah arus listrik searah atau DC. Motor DC terdiri dari komponen utama yaitu rotor dan stator, serta beberapa komponen pendukung seperti kutub medan magnet dan sikat arang.

Pada tugas akhir ini, motor DC Gearbox PG28 dipilih sebagai penggerak utama untuk sistem robot 3 roda omnidirectional, motor DC

Gearbox PG28 bisa dilihat pada gambar 2.7. Motor ini memiliki spesifikasi tegangan suplai DC 24V, arus 2A, kecepatan putaran 800 rpm, torsi 7 kg/cm, dan berat 250 gram. Dengan kombinasi kecepatan tinggi, torsi yang mencukupi, dan ukuran yang kompak, motor DC Gearbox PG28 menjadi solusi yang ideal untuk memberikan pergerakan akurat dan efisien pada robot[20].



Gambar 2. 8 Motor DC gearbox pg