

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada pengembangan *Trackless* AGV berbasis LiDAR, penulis meninjau informasi dari berbagai referensi berupa penelitian ilmiah terdahulu yang dilakukan oleh peneliti-peneliti berkompeten di bidang yang berkaitan dengan topik penelitian.

1. Penelitian Nadia Tri Jayanti, Angga Rusdinar, Ph.D, dan Agung Surya Wibowo, M.T. yang berjudul “*PERANCANGAN SISTEM PENGONTROLAN PERGERAKAN AUTOMATED GUIDED VEHICLE (AGV) UNTUK MENARIK TROLI MENGGUNAKAN SENSOR LIDAR*” (Jayanti, dkk.).

Di dalam penelitian ini dibahas mengenai perancangan AGV berbasis sensor Rplidar untuk mengangkat troli yang berisi barang. AGV yang dipakai di dalam penelitian ini adalah AGV yang berbasis *line follower*. AGV berbasis *line follower* diketahui memiliki beberapa kelemahan dan keterbatasan di dalam pengangkutan troli. Salah satunya adalah posisi AGV tidak selalu berada di koordinat yang benar akibat dari posisi troli yang tidak lurus dengan garis yang dilalui oleh AGV. Untuk mengatasi masalah tersebut, maka AGV tersebut dipasangi sensor LiDAR yang berfungsi untuk melakukan lokalisasi dan menentukan posisi robot secara otomatis.

Secara umum, penelitian ini memiliki tujuan yang sangat mirip dengan

proyek kami, yaitu merancang AGV yang memiliki sensor untuk mendeteksi

lokasi robot dan objek lain secara real time. Akan tetapi, di dalam penelitian ini sistem line following yang dimiliki AGV disandingkan dengan sensor LiDAR untuk mendeteksi lokasi troli. Sedangkan penelitian kami bertujuan untuk menggunakan sensor LiDAR secara utuh dan menghilangkan sistem *line following*.

2. Penelitian Amelia Emara, Angga Rusdinar, dan Ramadhan Nugraha yang berjudul “PERANCANGAN ALGORITMA SISTEM PENGHINDAR TABRAKAN PADA AUTOMATIC GUIDED VEHICLE (AGV) MENGGUNAKAN SENSOR LIDAR”.

Penelitian ini bertujuan untuk menambahkan kemampuan collision avoidance atau menghindari tabrakan di robot AGV menggunakan sensor Rplidar. Kemampuan collision avoidance memungkinkan AGV untuk menghindari tabrakan saat ada objek yang secara tiba-tiba berada di jalur lintasan gerak AGV. Cara kerja collision avoidance AGV di dalam penelitian

ini adalah AGV akan berhenti sejenak saat mendeteksi objek di depannya, kemudian menghitung jarak titik tengah objek dan lebar celah kanan dan kiri.

Setelah itu, AGV akan menghindar sesuai dengan lebar celah yang dapat dilalui AGV.

3. Penelitian dari Murat Köseoğlu, Orkan Murat Çelik, Ömer Pektaş yang berjudul “DESIGN OF AN AUTONOMOUS MOBILE ROBOT BASED ON ROS”.

Penelitian ini berfokus kepada perancangan dan pengembangan Autonomous Mobile Robot (AMR) yang berbasis Robot Operating System (ROS). ROS adalah middleware atau sekumpulan perangkat lunak yang menyediakan fungsionalitas untuk perangkat robot seperti mapping, lokalisasi dan navigasi. Hasilnya, robot yang dirancang dengan sistem ROS di dalam penelitian ini dapat melakukan berbagai tugas seperti pemetaan lingkungan, lokalisasi atau menentukan lokasi robot di dalam map, dan obstacle avoidance atau menghindari objek dan rintangan. Namun, satu hal yang penting untuk digaris bawahi adalah di dalam penelitian ini diketahui bahwa saat kecepatan robot ditingkatkan atau bertambah, maka sensitivitas dan akurasi sensor mapping dari robot berkurang.

2.2 Dasar Teori

2.2.1. Automated Guided Vehicle (AGV)

AGV adalah kendaraan pembawa muatan yang berjalan tanpa operator atau pengemudi. Gerakan AGV dipandu oleh kombinasi

perangkat lunak dan sistem panduan yang berbasis sensor. Aplikasi AGV yang umum mencakup transportasi bahan mentah, barang dalam proses, dan barang jadi untuk mendukung jalur produksi manufaktur, dan penyimpanan/pengambilan atau gerakan lain untuk mendukung pengambilan dalam aplikasi pergudangan dan distribusi.

2.2.2. Trackless AGV

Trackless AGV adalah jenis AGV yang dapat berjalan tanpa track atau lintasan khusus yang dijadikan panduan saat beroperasi. Umumnya, lintasan tersebut berupa garis yang melintang sepanjang wilayah kerja dan operasi dari AGV. Trackless AGV dapat berjalan dengan mandiri dengan bantuan sensor yang membantu untuk memetakan lingkungan dan mendeteksi objek. Sensor yang paling umum ditemukan di dalam trackless AGV adalah kamera dan LiDAR.

2.2.3. Light Detection And Ranging (LiDAR)

LiDAR adalah metode yang digunakan untuk mengukur jarak suatu objek dengan laser. Cara kerja LiDAR adalah menembakkan laser ke arah suatu objek dan menghitung waktu yang dibutuhkan bagi pantulan laser tersebut untuk kembali ke penerima (LiDAR). LiDAR biasanya digunakan untuk membuat model tiga dimensi dari sebuah area di permukaan bumi atau dasar lautan. Namun, LiDAR juga dapat digunakan untuk membuat peta dengan resolusi tinggi yang dapat diaplikasikan di bidang survei, geodesi, geomatika, arkeologi, geografi,

geologi, geomorfologi, seismologi, kehutanan, fisika atmosfer, panduan laser, pemetaan petak laser udara (ALSM), dan altimetri laser.

Di bidang robotik, LiDAR digunakan untuk mapping dan localization. Mapping adalah memetakan area atau lingkungan di sekitar robot, sedangkan localization adalah menentukan lokasi robot secara real time. Metode localization yang umum digunakan oleh LiDAR adalah Simultaneous Localization and Mapping atau SLAM.

2.2.4. Robot Operating System (ROS)

ROS adalah sekumpulan perangkat lunak untuk pengembangan robot. Ia menyediakan layanan yang dirancang untuk sekumpulan komputer yang heterogen seperti abstraksi perangkat keras, kontrol perangkat tingkat rendah, implementasi fungsionalitas yang umum digunakan, penyampaian pesan antar proses, dan manajemen paket. Proses ROS direpresentasikan sebagai node dalam struktur grafik, dihubungkan oleh yang disebut topik. Node ROS dapat mengirimkan pesan satu sama lain melalui topik, membuat panggilan layanan ke node lain, menyediakan layanan untuk node lain, atau mengatur atau mengambil data bersama dari database komunal yang disebut server parameter.

Seluruh proses ROS dikendalikan oleh sebuah proses yang disebut ROS Master. ROS Master memungkinkan semua ini dengan mendaftarkan node ke dirinya sendiri, menyiapkan komunikasi node-to-node untuk topik, dan mengontrol pembaruan server parameter. Pesan

dan panggilan layanan tidak melewati master, melainkan master mengatur komunikasi peer-to-peer antara semua proses node setelah mereka mendaftarkan diri dengan master. Arsitektur terdesentralisasi ini cocok untuk robot, yang sering kali terdiri dari subset perangkat keras komputer jaringan, dan dapat berkomunikasi dengan komputer off-board untuk komputasi atau perintah berat.

1. Node

Sebuah node mewakili satu proses yang menjalankan grafik ROS. Setiap node memiliki nama, yang didaftarkan dengan master ROS sebelum dapat mengambil tindakan lain. Beberapa node dengan nama yang berbeda dapat berada di bawah ruang nama yang berbeda, atau sebuah node dapat didefinisikan sebagai anonim, dalam hal ini secara acak akan menghasilkan pengenal tambahan untuk ditambahkan ke nama yang diberikan. Node berada di pusat pemrograman ROS, karena sebagian besar kode klien ROS berbentuk node ROS yang mengambil tindakan berdasarkan informasi yang diterima dari node lain, mengirim informasi ke node lain, atau mengirim dan menerima permintaan tindakan ke dan dari node lain.

2. Topic

Topik diberi nama bus di mana node mengirim dan menerima pesan. Nama topik juga harus unik di dalam namespace-nya. Untuk mengirim pesan ke suatu topik, sebuah node harus memublikasikan

ke topik tersebut, sedangkan untuk menerima pesan harus berlangganan. Model publish/subscribe bersifat anonim: tidak ada node yang mengetahui node mana yang mengirim atau menerima pada suatu topik, hanya node yang mengirim/menerima topik tersebut. Jenis pesan yang disampaikan pada suatu topik sangat bervariasi dan dapat ditentukan oleh pengguna. Isi dari pesan ini dapat berupa data sensor, perintah kontrol motor, informasi keadaan, perintah aktuator, atau lainnya.

3. Service

Sebuah node juga dapat mengiklankan layanan atau service. Layanan mewakili tindakan yang dapat diambil oleh simpul yang akan memiliki satu hasil. Dengan demikian, layanan sering digunakan untuk tindakan yang memiliki awal dan akhir yang ditentukan, seperti menangkap gambar satu bingkai, daripada memproses perintah kecepatan ke motor roda atau data odometer dari enkoder roda. Node mengiklankan layanan dan layanan panggilan dari satu sama lain.

4. Server Parameter

Server parameter adalah database yang digunakan bersama di antara node yang memungkinkan akses komunal ke informasi statis atau semi-statis. Data yang tidak sering berubah dan dengan demikian akan jarang diakses, seperti jarak antara dua titik tetap di

lingkungan, atau berat robot, adalah kandidat yang baik untuk penyimpanan di server parameter.

2.2.5. Simultaneous Localization and Mapping (SLAM)

SLAM adalah metode yang digunakan oleh kendaraan otonom yang untuk membuat peta dan melokalisasi kendaraan di peta itu di saat yang bersamaan. Algoritma SLAM memungkinkan kendaraan untuk memetakan lingkungan yang tidak diketahui. SLAM sangat berguna untuk melakukan tugas-tugas seperti perencanaan jalur dan penghindaran rintangan. SLAM juga berguna dalam banyak aplikasi lain seperti melakukan navigasi armada robot seluler untuk mengatur rak di gudang, memarkir mobil tanpa pengemudi di tempat kosong, atau mengirimkan paket dengan menavigasi drone di lingkungan yang asing atau belum dikenal sebelumnya.

Ada dua jenis teknologi yang terlibat di dalam SLAM. Teknologi pertama adalah sensor signal processing yang di dalamnya termasuk front-end processing yang sangat tergantung pada sensor yang digunakan. Teknologi kedua adalah pose-graph optimization yang di dalamnya termasuk pemrosesan back-end, yang merupakan sensor-agnostik.

2.2.6. RoslibJS

ROSLIBJS adalah sebuah library javascript yang menyediakan fungsionalitas untuk berinteraksi dengan layanan ROS dari browser.

Komunikasi dari browser ke layanan ROS difasilitasi oleh protokol websocket. Aplikasi yang menggunakan ROSLIBJS berfungsi sebagai websocket client, dan layanan ROS berfungsi sebagai websocket server dengan bantuan `rosbridge_suite`.

2.2.7. WebSocket

WebSocket adalah protokol komunikasi komputer yang menyediakan saluran komunikasi full-duplex melalui koneksi TCP. Protokol websocket memungkinkan interaksi antara browser (atau aplikasi klien lainnya) dan server dengan overhead yang lebih rendah daripada alternatif half-duplex seperti HTTP. WebSocket memfasilitasi transfer data secara real-time dari dan ke server dengan cara menyediakan metode bagi server untuk mengirim konten ke klien tanpa diminta terlebih dahulu oleh klien, dan memungkinkan pesan untuk diteruskan bolak-balik sambil menjaga koneksi tetap terbuka. Dengan cara ini, percakapan dua arah yang berkelanjutan dapat terjadi antara client dan server.