

2. TINJAUAN PUSTAKA

2.1. Teori Penunjang

Berikut merupakan sistem, metode, dan algoritma yang digunakan pada penelitian ini.

2.1.1. ROS (Robot Operating System)

ROS merupakan sistem operasi meta yang menghubungkan perangkat lunak (*operating system*) dengan perangkat keras yaitu sensor dan aktuator. ROS memiliki fitur *publisher* dan *subscriber* yang memungkinkan komunikasi antar *node* terjadi. Sehingga pengiriman pesan dari *node* untuk membaca sensor menuju *node* pemrosesan data dan menuju *node* untuk menjalankan aktuator dapat dilaksanakan di dalam ROS. Fitur pengiriman pesan juga dapat dilakukan antar perangkat dalam satu jaringan yang sama. Dikarenakan ROS bersifat *open-source* sehingga banyak *library* dan *package* yang dapat dimanfaatkan untuk suatu kebutuhan (Breiling et al., 2017; Estefo et al., 2019; Pyo et al., 2017).

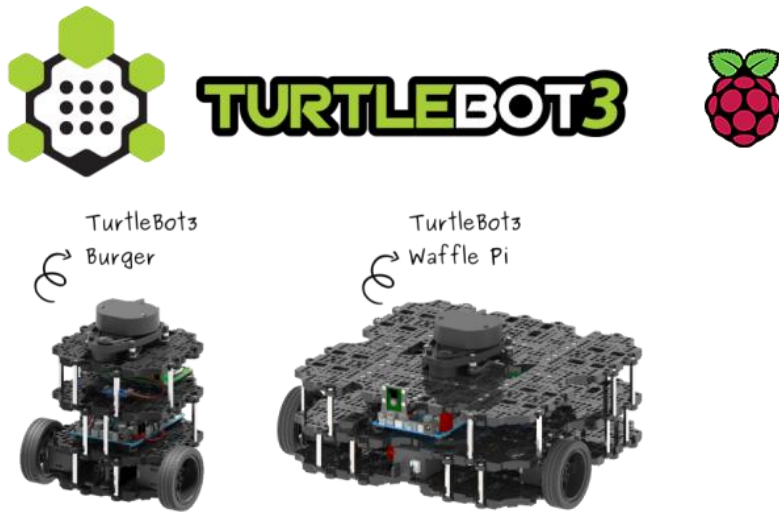


Gambar 2.1. Logo ROS

Sumber: (1200px-Ros_logo.Svg.Png (1200×319), n.d.)

2.1.2. Turtlebot3

TurtleBot adalah *platform* robot standar ROS. Turtle berasal dari Turtle robot, yang dijalankan oleh Logo bahasa pemrograman komputer untuk pendidikan pada tahun 1967. Selain itu, *node* turtlesim, yang pertama kali ada dalam tutorial dasar ROS, adalah program yang meniru sistem perintah dari Logo *turtle program*. Itu juga digunakan untuk membuat ikon Turtle sebagai simbol ROS. Sembilan titik yang digunakan dalam logo ROS berasal dari cangkang belakang kura-kura. TurtleBot, yang berasal dari Turtle dari Logo, dirancang untuk dengan mudah mengajar orang-orang yang baru mengenal ROS melalui TurtleBot serta untuk mengajar bahasa pemrograman komputer menggunakan Logo. Sejak itu TurtleBot telah menjadi platform standar ROS, yang merupakan platform paling populer di kalangan pengembang dan siswa.



Gambar 2.2. Robot Turtlebot3

Sumber: (TurtleBot3, n.d.)

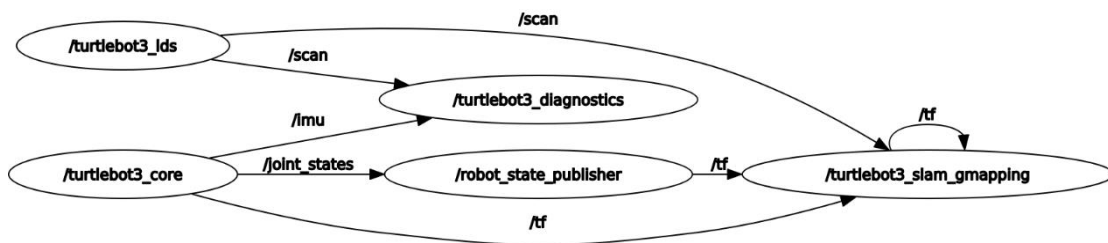
Ada 3 versi model TurtleBot. TurtleBot1 dikembangkan oleh Tully (Pengelola Platform di Open Robotics) dan Melonee (CEO Fetch Robotics) dari Willow Garage di atas robot penelitian berbasis Roomba iRobot, Create, untuk penyebaran ROS. Ini dikembangkan pada 2010 dan telah dijual sejak 2011. Pada 2012, TurtleBot2 dikembangkan oleh Yujin Robot berdasarkan robot penelitian, iCleo Kobuki. Pada tahun 2017, TurtleBot3 dikembangkan dengan fitur untuk melengkapi kekurangan fungsi pendahulunya, dan tuntutan pengguna. TurtleBot3 mengadopsi ROBOTIS *smart actuator* DYNAMIXEL untuk mengemudi.

TurtleBot3 adalah robot kecil, terjangkau, dapat diprogram, berbasis ROS untuk digunakan dalam pendidikan, penelitian, hobi, dan pembuatan prototipe produk. Tujuan dari TurtleBot3 adalah untuk secara dramatis mengurangi ukuran platform dan menurunkan harga tanpa harus mengorbankan fungsionalitas dan kualitasnya, sementara pada saat yang sama menawarkan perluasan. TurtleBot3 dapat dikustomisasi ke dalam berbagai cara tergantung pada bagaimana pengguna merekonstruksi bagian mekanis dan menggunakan bagian opsional seperti komputer dan sensor. Selain itu, TurtleBot3 dikembangkan dengan SBC yang hemat biaya dan berukuran kecil yang cocok untuk sistem tertanam yang kuat, sensor jarak 360 derajat, dan teknologi pencetakan 3D.

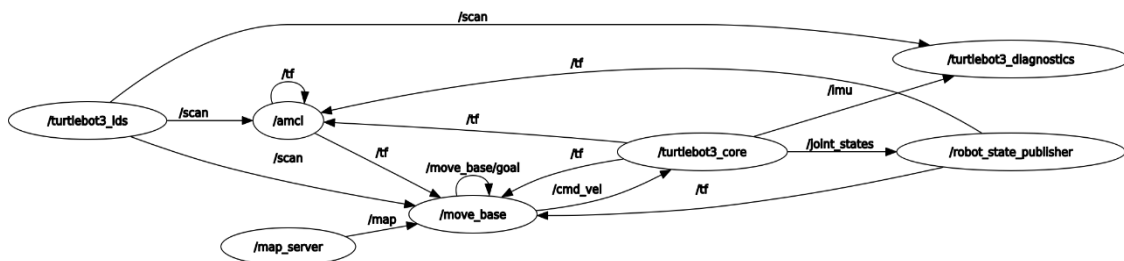
Teknologi inti TurtleBot3 adalah SLAM, Navigasi, dan Manipulasi, sehingga cocok untuk robot layanan rumah. TurtleBot dapat menjalankan algoritma SLAM untuk membuat peta dan dapat berkeliling ruangan Anda. Selain itu, dapat dikendalikan dari jarak jauh dari laptop, joypad, atau ponsel pintar berbasis Android. TurtleBot juga dapat mengikuti kaki seseorang saat mereka

berjalan di dalam ruangan. TurtleBot3 juga dapat digunakan sebagai manipulator seluler yang mampu memanipulasi objek dengan memasang manipulator seperti OpenMANIPULATOR. OpenMANIPULATOR memiliki keuntungan karena kompatibel dengan TurtleBot3 Waffle dan Waffle Pi. Melalui kompatibilitas ini dapat mengimbangi kurangnya kebebasan dan dapat memiliki kelengkapan yang lebih besar sebagai robot layanan dengan SLAM dan kemampuan navigasi yang dimiliki TurtleBot3. (TurtleBot3, n.d.)

Berikut merupakan diagram hubungan antar *node* yang dimiliki oleh Turtlebot3 ketika melakukan pemetaan dan navigasi. Setiap *node* memiliki fungsinya masing-masing yang saling terhubung dan berkomunikasi pada topik-topik tertentu.



Gambar 2.3. Diagram hubungan antar *node* saat pemetaan



Gambar 2.4. Diagram hubungan antar *node* saat navigasi

2.2. Tinjauan Sistem dari Penelitian yang Sudah Ada

Berikut merupakan tinjauan sistem dari penelitian yang sudah ada.

2.2.1. Penerapan Sistem Navigasi Mobile Robot pada Turtlebot dengan Depth Sensor Camera

Peneliti Aprilino pada tahun 2020 meneliti mengenai robot Turtlebot3 yang menggunakan Depth Sensor Camera. Platform Turtlebot3 yang digunakan memanfaatkan ROS sebagai dasar dalam pemrograman sistem *mobile robot*. Depth Sensor Camera yang terpasang digunakan untuk mendeteksi benda di sekitar letak robot bergerak. Pada penelitian ini dilakukan pengujian dengan menguji cara kerja Depth Sensor Camera dan mengambil data apa saja yang dapat digunakan dari sensor. Data dari sensor akan digunakan untuk membuat peta dan hasil dari peta yang telah terbuat akan dicocokkan dengan kondisi pada lingkungan sebenarnya. Penelitian ini mendapati bahwa robot dapat menuju titik yang diinginkan pada pengujian tanpa halangan dan menggunakan halangan. (Aprilino, 2020)

2.2.2. Design of an Autonomous Mobile Robot Based on ROS

Dalam penelitian ini, desain dari AMR yang diadaptasi untuk ROS disajikan dengan mempertimbangkan arsitektur perangkat keras dan protokol komunikasi elektronik. Awalnya tujuan robot telah ditentukan dan kemudian komponen yang diperlukan untuk membangun robot telah diperoleh. Setelah memasang komponen perangkat keras elektronik dan mekanik pada platform, interkoneksi yang diperlukan, sistem pertukaran data, dan perangkat lunak telah diinstal. Kemudian pengujian yang diperlukan telah dijalankan pada platform dan beberapa metode solusi telah diusulkan untuk masalah yang dihadapi selama pengujian. Hasil yang diperoleh dalam penelitian ini sudah sesuai tujuan. (Köseoğlu et al., 2017)

2.2.3. Path Planning of Autonomous Mobile Robot

Penelitian ini menyajikan algoritma untuk perencanaan jalur ke target untuk robot mobil di lingkungan yang tidak diketahui. Algoritma yang diusulkan memungkinkan robot bergerak untuk menavigasi melalui rintangan statis, dan menemukan jalan untuk mencapai target tanpa tabrakan. Algoritma ini memberikan robot kemungkinan untuk berpindah dari posisi awal ke posisi akhir (target). Strategi pencarian jalur yang diusulkan dirancang dalam bentuk peta-grid dari lingkungan yang tidak diketahui dengan hambatan statis yang tidak diketahui. Robot bergerak dalam lingkungan yang tidak diketahui dengan merasakan dan menghindari rintangan yang datang ke arah target. Ketika misi dijalankan, perlu untuk merencanakan jalur yang optimal

atau layak untuk dirinya sendiri menghindari penghalang di jalannya dan meminimalkan biaya seperti waktu, energi, dan jarak. Perencanaan jalur yang diusulkan harus membuat robot dapat mencapai tugas-tugas ini: menghindari rintangan, dan membuat jalan menuju sasarannya. Algoritma tersebut diimplementasikan dalam Borland C++, kemudian diuji dengan visual basic dan bahasa pemrograman DELPHI; dimana lingkungan dipelajari dalam sistem koordinat dua dimensi. Bagian simulasi adalah pendekatan terhadap hasil nyata yang diharapkan; bagian ini dilakukan dengan menggunakan C++ untuk mengenali semua objek dalam lingkungan dan karena itu cocok untuk masalah grafis. Mengambil lingkungan tersegmentasi yang dikeluarkan dari pengembangan C++, algoritma mengizinkan robot untuk berpindah dari posisi awal ke posisi yang diinginkan mengikuti perkiraan lintasan menggunakan Visual Basic dan bahasa Delphi. (Hachour, 2008)