

- a) ROS (Robot Operating System) adalah seperangkat library perangkat lunak dan peralatan untuk membantu pembuatan aplikasi robotik. ROS menyediakan lapisan abstraksi perangkat keras yang memungkinkan pembuatan aplikasi robotik tanpa didasari perangkat keras yang khusus. Selain itu, ROS juga dapat digunakan untuk menyediakan perangkat lunak yang dibutuhkan untuk visualisasi dan debug data robot. Pada esensinya, ROS adalah middleware yang dapat bertukar data satu sama lain bahkan saat beroperasi di mesin yang berbeda. Hal ini yang membuat ROS dapat mengintegrasikan komponen-komponen robot seperti sensor, aktuator, dan kamera dengan memungkinkan pertukaran antar komponen.
- b) ROS 2 menggunakan Data Distribution Service (DDS) yang lebih efisien, lebih dapat diandalkan, latency yang lebih rendah, dibandingkan ROS1 yang menggunakan ROS Master-Slave Architecture dan XML-RPC middleware. Selain itu, ROS2 memiliki parameter Quality of Service (QoS) yang dapat dikonfigurasi.

ROS memiliki 2 library yang terpisah, yaitu roscpp untuk C++ dan rospy untuk Python, sedangkan ROS 2 memiliki basis library dalam C, yang kemudian digunakan menjadi dasar pembuatan library-library berikutnya. Hal ini menyebabkan ROS2 mampu beroperasi dalam bahasa pemrograman yang lebih banyak.

Rosbag pada ROS2 memberikan lebih banyak fleksibilitas serealisasi dibandingkan ROS1 yang menggunakan serealisasinya sendiri.

ROS2 juga menyediakan fitur QoS (Quality of Service) yang memungkinkan konfigurasi aliran data, Multi threaded execution yang memungkinkan beberapa nodes berjalan pada waktu bersamaan, dan Real time processing.

- c) Menghemat waktu dan biaya.

Simulasi robotik memungkinkan pengujian fungsi robot tanpa harus mengeluarkan waktu dan uang untuk menyediakan hardware atau komponen fisik robot.

Meningkatkan keamanan

Simulasi memungkinkan pengujian fungsi robot tanpa resiko kecelakaan dan kerusakan.

Mempercepat pengembangan

Deteksi kesalahan secara dini dan perbaikan yang dilakukan dalam waktu cepat menghasilkan pengembangan robot yang semakin cepat.

Contoh simulasi yang dapat menghemat biaya dan waktu adalah pada simulasi sebuah robot yang diharapkan dapat bekerja di suhu tinggi. Dari segi biaya, pembuat robot tidak perlu mengeluarkan biaya untuk membeli hardware dan mencari lingkungan pengujian

yang sesuai. Dari segi waktu, pembuat robot tidak perlu menghabiskan waktu untuk mencari lingkungan yang sesuai.

- d) Gazebo adalah simulator yang dapat mensimulasikan kondisi fisik yang realistis dengan hukum-hukum fisika yang relevan dengan dunia nyata. Gazebo dapat digunakan bersama ROS (Robot Operating System) untuk melakukan simulasi algoritma kontrol, pathfinding, dan interaksi robot sebelum dilakukan secara nyata.

Langkah:

- 1) Mengunduh ROS dan Gazebo dan paket seperti gazebo\_ros dan ros\_control
  - 2) Membuat deskripsi robot dalam file URDF (Unified Robot Description Format)
  - 3) Membuat file peluncuran untuk meluncurkan Gazebo
  - 4) Menggunakan ros\_control untuk mengatur motor dan aktuator yang berhubungan
  - 5) Meluncurkan launch file yang telah dibuat
  - 6) Memberikan perintah ke robot dengan input manual
  - 7) Menggunakan Rviz untuk melakukan visualisasi data sensor
  - 8) Memonitor perintah yang sudah diberikan dan mengevaluasi respon robot
  - 9) \
- e) Untuk melakukan navigasi robot diperlukan beberapa hal berikut:
- 1) Mapping: Memanfaatkan sensor untuk membangun peta di lingkungan sekitar
  - 2) Localization: Penentuan lokasi robot pada peta
  - 3) Path Planning: Menemukan jalan paling optimal untuk mencapai tujuan
  - 4) Obstacle avoiding: Menghindari hambatan untuk menemukan jalan yang paling optimal
  - 5) Control system: Memastikan pada implementasinya robot mengikuti jalur yang sudah direncanakan sebelumnya.
- f) TF(transform) adalah sebuah library yang berfungsi untuk pelacakan bingkai koordinat komponen-komponen robotika dalam ruang tiga dimensi. TF memberikan informasi posisi robot dengan frame of reference sehingga komponen-komponen robot dapat merespon informasi yang diberikan tersebut dan dapat terjadi navigasi dan gerak robot yang benar.