

- a. ROS atau Robot Operating System adalah sebuah framework open-source yang digunakan untuk mendesain dan membangun sebuah software untuk sebuah robot. Software tersebut dapat berisi instruksi-instruksi yang perlu dijalankan oleh hardware dari robot. ROS sendiri sebenarnya bukan sebuah operating system (OS), tetapi merupakan sebuah middleware atau kumpulan tools dan libraries yang dijalankan pada sebuah OS konvensional, seperti Linux. ROS merupakan hal yang esensial dalam pengembangan sebuah robot karena memungkinkan interaksi dan integrasi antara berbagai komponennya. Hal ini dicapai dengan memberikan komponen-komponen sebuah framework komunikasi standar yang memanfaatkan nodes. Masing-masing setiap node dapat menangani pekerjaan suatu komponen.
- b. ROS2 merupakan pengembangan dari ROS pertama. Terdapat berbagai perbedaan utama yang menjadikan ROS2 sebagai pilihan bagi pengembang. Pertama adalah arsitektur middleware kedua ROS. ROS menggunakan Master-Slave Architecture yang memanfaatkan sistem sentral yang menangani komunikasi antara node. Sistem sentral ini menjadi kelemahan bagi ROS karena hanya memerlukan satu kelalaian untuk menjatuhkan seluruh software. Di sisi lain, ROS2 memanfaatkan Data Distribution Service (DDS) yang lebih fleksibel dan decentralized. Selain itu, DDS juga memiliki kelebihan secara efisiensi, kecepatan, dan latensinya yang rendah. DDS juga mendukung penyetelan beberapa pengaturan Quality of Service yang memungkinkannya untuk mendukung sistem yang real-time. Dalam segi keamanan, ROS2 dilengkapi dengan fitur-fitur yang belum diimplementasikan pada ROS, seperti enkripsi pesan antara node dan otentikasi node. Dalam segi pemeliharaan, ROS2 sebagai pengembangan terbaru tentu akan menerima berbagai fitur baru, perbaikan sistem, dll. ROS2 juga dapat berintegrasi dengan berbagai teknologi yang lebih baru. Di sisi lainnya, ROS yang sudah lebih tua akan semakin obsolete kedepannya dan berhenti menerima update sistem.
- c. Simulasi robotik merupakan tahap esensial dalam pengembangan robot karena memberikan gambaran yang cukup akurat kepada pengembang sebelum membangun robot fisik. Simulasi dapat dilakukan menggunakan berbagai platform atau software, seperti Gazebo, untuk memodelkan robot dan mensimulasikan gerakan robot. Simulasi juga penting karena dapat menghemat waktu dan biaya pengembang. Contohnya adalah saat sekelompok pengembang sedang memikirkan pengembangan suatu robot. Setelah mendesain robot dengan berbagai komponennya, alih-alih langsung mulai membeli bahan dan komponen robot, kelompok tersebut mencoba untuk memodelkan dan mensimulasikan robot terlebih dahulu. Setelah melakukan beberapa uji coba pada simulasi, mereka menemukan bahwa beberapa kelalaian dalam desain komponen penggerakannya dan sensornya. Alhasil, mereka mengubah-ubah desain awal dan menjalankan simulasi kembali dengan desain baru. Setelah berbagai uji coba yang berhasil, mereka pun mulai membangunnya secara fisik dan hasil robot fisiknya memuaskan bagi mereka.
- d. Gazebo adalah sebuah platform simulasi untuk mendesain dan memodelkan suatu robot serta mensimulasikan pergerakan fisik robot. Model robot umumnya

menggunakan format URDF (Unified Robot Description Format) atau SDF (Simulation Description Format), kemudian Gazebo memanfaatkan engine fisiknya untuk melakukan berbagai simulasi pada robot. Integrasi ROS dengan Gazebo dilakukan dalam berbagai tahap:

1. Penyetelan ROS
Untuk memulai integrasi, ROS perlu disetel terlebih dahulu. Hal ini dilakukan dengan membuat workspace yang baru untuk membuat software robot.
 2. Membuat model robot
Selanjutnya adalah mendesain dan membuat model robot. Seperti yang telah disebutkan, model robot dibuat dengan format URDF atau SDF.
 3. Menghubungkan ROS dengan Gazebo
ROS dan Gazebo dapat dihubungkan menggunakan sebuah plugin. Terdapat beberapa pilihan plugin yang dapat digunakan untuk mengendalikan komponen-komponen spesifik model robot
 4. Membuat dan menjalankan launch file
Berikutnya ialah membuat sebuah launch file. File tersebut menginisialisasi node, simulasi Gazebo, dan kontrol ROS. Setelah dijalankan, file tersebut akan memasukkan model robot ke dalam sebuah ruang berdimensi tiga.
 5. Mengendalikan robot
Terakhir, model robot yang dimasukkan ke dalam “dunia” digital dapat dikenalkan untuk melakukan berbagai macam pengujian.
 6. Navigasi model robot di dunia simulasi kerap menggunakan konsep pemetaan dan lokalisasi. Pemetaan merupakan sebuah proses yang dilakukan oleh robot untuk memetakan lingkungan di sekitarnya. Objek-objek di sekitar, seperti benda, dinding, lantai, dan lainnya dipetakan oleh robot. Di sisi lainnya, lokalisasi adalah proses robot menentukan posisi dan orientasinya pada lingkungan yang sudah dipetakan. Kedua fitur ini dapat diimplementasikan menggunakan teknik SLAM (Simultaneous Localization and Mapping). Sistem yang memanfaatkan metode SLAM umumnya memiliki dua tahap dalam prosesnya. Pertama adalah melakukan pengukuran lingkungan di sekitar robot. Hal ini dapat dicapai dengan sensor, laser LiDAR, ataupun sonar. Hasil-hasil pengukuran akan disimpan sebagai data yang akan diproses pada tahap berikutnya, yakni pemrosesan data. Pemrosesan ini menggunakan software untuk menginterpretasi data jarak agar robot mampu mengidentifikasi objek-objek di sekitarnya.
- e. TF adalah sebuah package atau framework yang memungkinkan pengguna untuk melacak berbagai kerangka koordinat atau frame pada model robot. TF mengelola posisi dan orientasi robot dalam bentuk translasi dan rotasi. Selain itu, TF juga dapat digunakan untuk mentransformasikan titik, vektor, dan lainnya antara dua frame pada model robot. Dengan begitu, robot dapat memahami posisi dan orientasinya pada peta lingkungan sekitarnya.
- TF dimanfaatkan oleh robot agar dipastikan dapat bergerak benar melalui beberapa cara. Pertama adalah dalam navigasi robot. TF dapat membantu robot dalam

melakukan pemetaan dengan mengubah data sensor ke dalam map, dan TF membantu lokalisasi robot dengan cara yang serupa juga dengan teknik SLAM. Kemudian, TF dapat mensimulasikan pergerakan robot dengan melakukan transformasi antara frame berupa translasi atau rotasi. Karena transformasi diperbarui secara real-time, model robot dapat mengetahui posisi komponen-komponennya dengan cepat dan mudah juga. Selanjutnya, pengguna dapat mengutak-atik gerakan transformasi untuk memastikan bahwa robot dapat bergerak benar dalam simulasi.