

- a. Robot Operating System (ROS) adalah sebuah framework open-source yang membantu para developer dan peneliti robotika membangun dan berbagi alat-alat antara aplikasi robotik yang berbeda. ROS menyediakan berbagai media untuk membantu developer membangun robot mereka, seperti libraries, kode, dan pengetahuan yang berasal dari komunitas mereka. Selama 10 tahun lebih, komunitas ROS telah berkembang dan membuat sebuah ekosistem yang membantu jutaan developer dan user yang berkontribusi dalam mengembangkan software ROS itu sendiri dan dunia robotika secara umum. ROS digunakan di banyak industri yang beragam dari bidang pertanian hingga medis dan digunakan oleh banyak perusahaan global seperti Otto Motors, Boston Dynamics, Airbus, Intel, dan lain-lain.
- b. ROS2 diciptakan sebagai sebuah perkembangan dari ROS untuk mengikuti perkembangan-perkembangan di dalam dunia robotika sejak ROS pertama kali diciptakan. Developer cenderung lebih memilih ROS2 dibandingkan dengan ROS karena memiliki fitur-fitur yang lebih luas dan sesuai dengan standar modern ketika dibandingkan dengan pendahulunya. Beberapa perbedaan ROS dan ROS2 adalah sebagai berikut:

Perbedaan	ROS	ROS2
API	ROS memiliki 2 library dasar yaitu roscpp untuk C++ dan rospy untuk Python. Kedua library ini dibangun secara independent dan terpisah sehingga API yang dimiliki kedua library tersebut tidak sama dan menyebabkan adanya perbedaan level perkembangan fitur.	ROS2 hanya memiliki satu library dasar yaitu rcl yang menjadi sebuah fondasi yang mengandung semua fitur penting ROS2. Hal ini menyebabkan lebih selarasnya library rclcpp (C++) dan rclpy (Python) dibandingkan dengan roscpp dan rospy
OS Support	Ubuntu	Ubuntu, MacOS, Windows 10
Middleware	Menggunakan middleware ROS Master-Slave Architecture dan XML-RPC yang unggul dalam remote procedure calls yang sederhana.	Menggunakan Data Distribution Service (DDS) yang didesain untuk menyediakan efficiency dan reliability yang tinggi. Karena kompleksitas yang lebih tinggi, DDS menyediakan support yang lebih unggul untuk real-time systems.

Dibandingkan dengan ROS, ROS2 dapat menangani lebih banyak robot tanpa adanya penurunan performa yang signifikan karena digunakannya DDS sebagai infrastruktur komunikasi. DDS juga menyediakan keamanan yang lebih baik untuk pengguna, sesuatu yang kurang di ROS karena tidak dipertimbangkannya untuk kegunaan komersial saat pertama kali diciptakan.

ROS akan menghentikan support pada tahun 2025, sehingga tidak akan lagi dikembangkan dan dipelihara setelah periode tersebut. Namun ROS2 akan berlanjut untuk dikembangkan dan dipelihara.

- c. Simulasi merupakan langkah yang penting dalam membuat sebuah robot. Simulasi bertujuan untuk pengujian dan memastikan robot benar-benar bekerja sebelum dibangun secara fisik. Hal ini mengakibatkan berkurangnya penghabisan sumber daya dan waktu.

Misalkan, jika kita ingin membuat sebuah robot menggunakan Raspberry Pi, sebaiknya dilakukan simulasi terlebih dahulu di aplikasi Tinkercad. Hal ini akan memastikan bahan-bahan yang kita butuh untuk membuat robot tersebut dan konfigurasi yang tepat sehingga tidak perlu membuang uang dan waktu untuk membangun konfigurasi robot yang salah.

- d. Gazebo merupakan sebuah software open-source yang berfungsi sebagai simulator 3D untuk mensimulasikan robot-robot dalam lingkungan indoor dan outdoor yang kompleks. Gazebo memiliki kemampuan untuk membangun sebuah lingkungan fisik untuk robot dan mengaplikasikan hukum-hukum fisika dari dunia nyata ke dalam simulasi. Gazebo juga dapat mensimulasikan sensor secara akurat dan dapat mengobservasi dan berinteraksi dengan lingkungan di dunia simulasi.

Gazebo dapat mengintegrasikan robot yang dibuat di ROS. Plugin `ros2_control` yang berada di Gazebo membiarkan kita mengendalikan actuator robot dengan input command yang kita berikan.

- e. Navigasi di simulasi Gazebo dapat dilakukan dengan menggunakan ROS Navigation Stack. ROS Navigation Stack memungkinkan robot di simulasi untuk melakukan lokalisasi (mengetahui lokasi robot saat ini) dan mapping (menggambarkan sebuah peta di sekitar robot).
- f. Transforms atau TF menyatakan posisi dan orientasi dari sebuah benda terhadap obyek benda. Di robotika, sebuah frame merupakan sistem koordinat yang mendeskripsikan posisi di axis x, y, dan z serta orientasi benda tersebut. Transforms merupakan nilai translasi dan rotasi yang dibutuhkan untuk mengubah frame awal menjadi frame tujuan. Di ROS, transforms dapat digunakan untuk menentukan posisi robot atau komponen dari robot dan memastikan bahwa robot bergerak dengan benar, dengan cara mengikuti nilai transforms.