

## Nomor 1 URO TEST

A. ROS (Robot Operating System) adalah middleware yang berorientasi robotik. Middleware disini merujuk kepada layer penghubung antara operating system dengan aplikasi yang terdapat di dalamnya. Sehingga ROS tidak bisa dikatakan sebagai sebuah operating system seperti windows dan linux, melainkan ROS berisikan kumpulan alat dan library yang dibuat dengan tujuan utama menjalankan sebuah robot.

ROS berperan utama dalam pengembangan robot dengan memberikan sebuah wadah bagi para developer untuk berinovasi dan membagikan inovasi inovasinya yang dapat dengan mudah diimplementasikan oleh developer, hal ini dikarenakan ROS menerapkan standarisasi dalam robotics development.

B. Pada intinya, ROS2 adalah tahapan selanjutnya dari development ROS. ROS dulu didesain pada awalnya untuk robot robot yang dibanding dengan sekarang, masih sederhana. Melihat perkembangan dunia komputasi yang sangat cepat, ROS2 diharapkan dapat mengadaptasi ROS terhadap landscape dunia komputasi yang cepat berkembang tersebut, dan hal ini tampak dari beberapa perbedaan utama yang ada di ROS:

### 1. Performa

ROS menggunakan XML-RPC sedangkan ROS2 menggunakan DDS (Data Distribution Service) architecture. DDS ini memungkinkan ROS2 untuk lebih efisien, lebih reliable, lebih gampang dicustomize, dan dengan latency yang lebih rendah. ROS2 juga berisikan fitur fitur yang krusial yang tidak ada di ROS seperti multi thread execution yang dapat memaksimalkan penggunaan cpu cpu bercore banyak di jaman sekarang, konfigurasi data flow, dan lain lain yang pada akhirnya meningkatkan performa dan paling besar berdampak dalam real time processing, yang ketika dibandingkan sama ROS, jauh lebih reliable dan lebih cepat.

### 2. Security

ROS2 lebih secure dikarenakan DDS tadi dibanding dengan ROS. Hal ini dikarenakan DDS memberikan kemungkinan untuk melakukan koneksi peer 2 peer dibanding dengan ROS yang membutuhkan server utama (ROS Master).

Hal ini memiliki beberapa dampak seperti misal ketika di ROS, setiap node harus melakukan koneksi kepada ROS Master, sehingga harus menggunakan koneksi yang tidak aman seperti melalui internet, sedangkan di ROS 2 tidak harus menghubungi server utama. Sifatnya yang peer 2 peer juga memastikan bahwa jika ada satu sistem yang rusak, komunikasi antar node yang tidak rusak masih tetap jalan, namun ketika di ROS, ROS Masternya rusak maka semua node tidak bisa berkomunikasi lagi. Ujung ujungnya ROS2 lebih aman, reliable dibanding ROS.

### 3. Pemeliharaan Jangka Panjang

ROS2 tentunya akan terus memiliki software support dari developernya seperti update update yang terus berjalan, sedangkan ROS akan memiliki EOL di 2025.

Ketiga aspek inilah yang membuat developer cenderung memilih ROS2

C. Karena melalui simulasi tentu kita dapat melihat bagaimana robot kita berjalan sebelum mengeluarkan modal dalam bentuk biaya dan waktu dalam membangun robot kita. Hal ini bisa membuat kita untuk menguji potensi potensi kegagalan dan membantu proses optimalisasi tanpa mengeluarkan biaya untuk membuat beberapa prototype dan modifikasi.

Misal ketika membuat sebuah robot penyelam laut dalam, ketika dilakukan tanpa simulasi maka akan sangat boros untuk melakukan testing. Bayangkan effort yang dibutuhkan untuk menyelam kedalaman laut, banyak yang harus dites sehingga robot aman menahan tekanan air di dalam dan koneksi sinyal dapat dipertahankan dengan baik agar tidak lost contact. Ketika robot turun dan rusak akibat tekanan air, maka prototipe baru harus dirancang lagi, padahal jika melalui simulasi hanya tinggal memencet tombol reset doang. Maka simulasi tentu sangat penting.

D. Gazebo adalah aplikasi yang dapat mensimulasi jalan kerja robot. Di dalam simulasi ini terdapat lingkungan fisik sehingga jalan kerja robot dapat dites dalam berbagai kondisi riil.

Berikut langkah langkah dasar mengintegrasikan ROS dengan Gazebo:

1. Install Gazebo dan ROS kedalam komputer
2. Jalankan Gazebo yang memiliki integrasi dengan ROS
3. Upgrade file URDF yang digunakan oleh ROS menjadi SDF yang dapat digunakan oleh Gazebo. Alat untuk mengkonversi ini sudah tersedia di aplikasi Gazebo sendiri
4. Lakukan modifikasi world dan robot yang ada di Gazebo dengan ROS dan simulasi berjalan dengan lancar

E. Robot secara general menavigasi lingkungannya melalui proses proses berikut:

1. Self localization

Self localization adalah proses dimana robot tersebut melalui poin poin data yang diterimanya (Seperti melalui computer vision) mencari tau posisi dirinya terhadap lingkungan sekitarnya.

2. Mapping

Dibaringi dengan mapping, dimana sebuah robot melalui poin poin data yang dimilikinya menciptakan sebuah sketsa terhadap lingkungannya yang dapat dipahaminya, sketsa ini bisa berisikan data data lainnya seperti point of interest, peta suhu, dan hal hal lainnya.

3. Pathing

Pathing merujuk kepada proses pengambilan langkah oleh robot. Pengambilan langkah ini didasari oleh data data yang diterima melalui mapping dan self localization.

F. Transform dalam ROS merujuk kepada sebuah package dalam ROS yang biasa disebut tf. tf memungkinkan developer untuk keep track beberapa poin koordinat yang bisa diubah ubah seiring dengan berjalannya waktu. Melalui package ini para developer dapat memastikan bahwa robot yang ingin digerakkan dapat bergerak dengan benar sesuai dengan data yang benar benar realtime sesuai realita yang ada. Misal ketika ada robot SAR yang menyusuri sebuah gua, proses mapping dan self localization sudah dilakukannya. Target dia adalah untuk mencapai target yang ingin ditolongnya, targetnya ini cukup jauh di depannya. Maka dia harus berjalan ke depan, memastikan tidak ada obstruksi baru dan ancaman ancaman yang terdapat seperti lubang yang ada berada di dekatnya atau batu batuan yang berjatuh. Proses pengecekan real time terhadap perubahan posisi posisi benda seperti dirinya dan lingkungan sekitarnya dapat dilakukan robot tersebut dengan tf.