

# Laporan Hasil Analisis Simulasi Robot Operating System (ROS)

## Chapter 2. Getting Started with ROS Programming

Robot Operating System (ROS) adalah kerangka kerja perangkat lunak yang digunakan untuk pengembangan perangkat lunak robot. ROS menyediakan berbagai alat dan pustaka untuk membantu proses pengembangan, serta manajemen perangkat keras robot. Chapter ini berisi seputar analisis konsep dasar tentang ROS, cara memulai pemrograman menggunakan ROS, dan langkah-langkah simulasi yang dilakukan untuk memverifikasi pengoperasian sistem. Selain itu, akan dijelaskan pula hasil yang diperoleh dari simulasi yang dilakukan.

### 1. Pengantar ROS

ROS merupakan sebuah framework open-source yang memberikan alat bantu bagi para pengembang untuk membangun sistem perangkat lunak robot secara lebih efisien dan modular. ROS menyediakan berbagai pustaka dan paket yang memungkinkan pengembang untuk mengakses berbagai fitur robotik, seperti kontrol motor, pengolahan citra, serta komunikasi antar robot.

ROS bukanlah sistem operasi secara langsung, melainkan lebih kepada kumpulan pustaka dan alat yang bekerja di atas sistem operasi Linux. ROS mendukung berbagai bahasa pemrograman, seperti Python dan C++, dan memungkinkan pengembangan aplikasi robot secara modular dengan berbasis pada konsep *node*, *topics*, *services*, dan *actions*.

### 2. Komponen Dasar ROS

ROS terdiri dari beberapa komponen utama yang sangat penting dalam pemrograman robot. Komponen-komponen ini meliputi:

- **Node:** Merupakan unit eksekusi dalam ROS yang melakukan tugas tertentu, seperti pemrosesan data atau kontrol perangkat keras.
- **Topic:** Sebuah saluran komunikasi yang memungkinkan pertukaran data antar node. Node dapat mengirim atau menerima data melalui topic.
- **Service:** Menyediakan komunikasi sinkron antara node. Berbeda dengan topic yang bersifat asinkron, service memungkinkan komunikasi request-response.
- **Parameter Server:** Tempat penyimpanan parameter yang dapat diakses oleh berbagai node dalam ROS.
- **Launch File:** Digunakan untuk menjalankan beberapa node sekaligus dengan konfigurasi tertentu.

### 3. Persiapan Sistem

Sebelum memulai pengembangan menggunakan ROS, ada beberapa langkah persiapan yang harus dilakukan:

- **Instalasi ROS:** Pada umumnya, ROS diinstal pada sistem operasi Ubuntu. Untuk instalasi, kita bisa menggunakan perintah `sudo apt-get install ros-<versi-ros>-desktop-full`.
- **Konfigurasi Workspace:** ROS bekerja dengan menggunakan workspace, yaitu direktori tempat pengembangan dilakukan. Workspace dapat dibuat dengan perintah `catkin init` dan kemudian menjalankan `catkin build` untuk membangun workspace.
- **Sumber Daya dan Dependensi:** ROS membutuhkan pustaka dan paket tambahan untuk berbagai aplikasi. Pustaka ini dapat diinstal menggunakan perintah `rosdep install`.

## 4. Langkah-langkah Simulasi

Setelah menyiapkan lingkungan pengembangan, langkah selanjutnya adalah melakukan simulasi untuk memverifikasi apakah sistem berjalan dengan baik.

### Langkah 1: Membuat Paket ROS

Untuk memulai, buatlah paket ROS yang berisi kode program untuk simulasi. Berikut adalah langkah-langkah untuk membuat paket baru:

- Jalankan perintah `catkin_create_pkg <nama_paket> <dependencies>`, di mana `<nama_paket>` adalah nama paket yang akan dibuat, dan `<dependencies>` adalah paket lain yang dibutuhkan (misalnya, `rospy` untuk Python).

### Langkah 2: Membuat Node Pertama

Setelah paket dibuat, buatlah sebuah node sederhana untuk mengirim pesan ke topik. Berikut adalah contoh kode Python untuk membuat node yang mempublikasikan pesan ke topik. Node ini akan mempublikasikan pesan "Hello ROS" ke topik bernama `chatter`.

### Langkah 3: Membuat Node Subscriber

Selanjutnya, buatlah node yang berfungsi untuk menerima pesan dari topik `chatter` yang dipublikasikan oleh node sebelumnya. Berikut adalah contoh kode Python untuk membuat node subscriber. Node ini akan mendengarkan topik `chatter` dan mencetak pesan yang diterima.

### Langkah 4: Menjalankan Simulasi

Setelah membuat node-node tersebut, langkah selanjutnya adalah menjalankan simulasi di terminal.

1. Pertama, buka terminal dan jalankan perintah `roscore` untuk memulai master ROS. Kemudian, buka terminal baru dan jalankan node penerbit dengan perintah:  
`roslaunch <nama_paket> talker.py`

2. Kemudian, buka terminal baru dan jalankan node penerbit dengan perintah: `roslaunch <nama_paket> talker.py`
3. Di terminal lain, jalankan node pendengar dengan perintah:  
`roslaunch <nama_paket> listener.py`

### Langkah 5: Verifikasi Hasil

Setelah kedua node dijalankan, node pendengar akan menerima pesan dari node penerbit dan mencetak "I heard Hello ROS" di terminal. Ini menunjukkan bahwa komunikasi antara node berjalan dengan baik melalui topik yang telah dibuat.

## 5. Hasil yang Diperoleh

Setelah mengikuti langkah-langkah di atas, hasil yang diperoleh adalah sebagai berikut:

1. Node penerbit berhasil mempublikasikan pesan ke topik `chatter`.
2. Node pendengar berhasil menerima dan menampilkan pesan dari node penerbit.
3. Proses komunikasi antara node menggunakan sistem topik berjalan sesuai harapan, yang menunjukkan bahwa ROS berhasil mengelola komunikasi antar node secara asinkron.

## 6. Kesimpulan

Pada chapter ini, telah dilakukan analisis tentang cara memulai dengan pemrograman ROS, termasuk pembuatan paket, pengembangan node, dan simulasi komunikasi antara node menggunakan topik. Proses ini mengkonfirmasi bahwa ROS menyediakan kerangka kerja yang sangat berguna dalam pengembangan aplikasi robotik dengan mendukung komunikasi antar node, serta pengelolaan sistem secara modular. Pemrograman ROS menggunakan bahasa Python mempermudah pengembang dalam memanfaatkan pustaka dan paket yang telah disediakan oleh ROS.

## Chapter 3. Working with ROS for 3D Modeling

Pada chapter ini, akan dibahas cara ROS digunakan untuk membangun dan memanipulasi model 3D yang diperlukan dalam aplikasi robotik. ROS mendukung berbagai alat dan pustaka yang memungkinkan pengembangan model 3D dalam konteks robotika, seperti simulasi lingkungan robot dan pemrograman visual. Dalam chapter ini, kita akan menganalisis cara menggunakan ROS untuk berinteraksi dengan model 3D, memvisualisasikan data dari sensor 3D, dan menjalankan simulasi untuk memverifikasi hasil yang diperoleh.

### 1. Pengantar ROS untuk Pemodelan 3D

ROS menyediakan berbagai alat yang memungkinkan pengguna untuk membuat, mengedit, dan mengelola model 3D untuk robot dan lingkungan sekitarnya. Beberapa fitur yang digunakan dalam pemodelan 3D di ROS meliputi:

- **Gazebo:** Sebuah simulator robot yang digunakan untuk simulasi robot dalam lingkungan 3D.
- **RViz:** Sebuah alat visualisasi untuk data sensor dan model 3D robot.
- **PCL (Point Cloud Library):** Pustaka yang digunakan untuk memproses data cloud titik dari sensor 3D, seperti LiDAR atau kamera depth.

Pada umumnya, pemodelan 3D di ROS sangat berguna dalam simulasi robot dan pengujian algoritma pemrograman sebelum diterapkan pada perangkat keras yang sesungguhnya.

### 2. Komponen Dasar untuk Pemodelan 3D di ROS

Untuk bekerja dengan model 3D di ROS, ada beberapa komponen dan alat yang penting untuk dipahami:

- **Gazebo:** Gazebo adalah simulator 3D yang kuat yang dapat digunakan untuk menguji robot dalam berbagai kondisi. ROS dapat berintegrasi dengan Gazebo untuk mensimulasikan robot dan lingkungan 3D.
- **RViz:** RViz adalah alat visualisasi 3D untuk ROS yang memungkinkan kita untuk melihat data sensor dalam bentuk visual, seperti data dari LiDAR atau kamera depth.
- **PCL (Point Cloud Library):** PCL digunakan untuk mengelola dan menganalisis data 3D yang diperoleh dari sensor. ROS mendukung penggunaan PCL untuk memproses data point cloud untuk aplikasi seperti pemetaan dan pemrograman pergerakan robot.

### 3. Persiapan Sistem

Sebelum memulai pemodelan 3D dengan ROS, ada beberapa langkah yang harus dipersiapkan:

- **Instalasi ROS:** ROS dapat diinstal pada sistem operasi Ubuntu dengan perintah `sudo apt-get install ros-<versi-ros>-desktop-full`. Selain itu, untuk mendukung pemodelan 3D, Gazebo dan RViz harus terinstal.

- **Instalasi Gazebo:** Untuk instalasi Gazebo, kita dapat menggunakan perintah `sudo apt-get install gazebo11 ros-<versi-ros>-gazebo-ros-pkgs`.
- **Instalasi PCL:** PCL dapat diinstal dengan perintah `sudo apt-get install libpcl-dev`.
- **Konfigurasi Workspace:** Workspace ROS harus disiapkan dengan menggunakan `catkin init` dan kemudian menjalankan `catkin build`.

## 4. Langkah-langkah Simulasi

Setelah sistem disiapkan, kita dapat memulai dengan melakukan simulasi yang melibatkan pemodelan 3D. Berikut adalah langkah-langkah simulasi yang akan dilakukan.

### Langkah 1: Membuat Paket ROS untuk Pemodelan 3D

Langkah pertama adalah membuat paket ROS baru yang akan berisi kode program untuk melakukan simulasi model 3D dengan menggunakan perintah `catkin_create_pkg <nama_paket> gazebo_ros rospy`. Paket ini akan bergantung pada pustaka Gazebo dan ROS, yang diperlukan untuk simulasi robot di lingkungan 3D.

### Langkah 2: Mengatur Model 3D Robot

Salah satu elemen utama dalam pemodelan 3D adalah robot itu sendiri. Misalnya, kita dapat menggunakan model robot sederhana yang telah disediakan oleh ROS atau membuat model robot kita sendiri menggunakan format URDF (Unified Robot Description Format). Sebagai contoh, kita dapat menggunakan model robot TurtleBot yang disediakan oleh ROS dengan menggunakan perintah `sudo apt-get install ros-<versi-ros>-turtlebot`. Kemudian muat model robot ke dalam Gazebo dengan menggunakan file URDF yang berisi deskripsi robot. File ini mendefinisikan bagian-bagian robot, seperti roda, sensor, dan chasis.

### Langkah 3: Memulai Gazebo dan RViz

Setelah model robot disiapkan, kita dapat memulai Gazebo untuk mensimulasikan robot dalam lingkungan 3D dengan menggunakan perintah `roslaunch turtlebot_gazebo turtlebot_world.launch`. Perintah ini akan membuka Gazebo dengan model TurtleBot yang akan bergerak di lingkungan virtual. Pada saat yang sama, kita dapat memvisualisasikan data sensor robot di RViz dengan menggunakan perintah `roslaunch turtlebot_rviz_launchers view_navigation.launch`. RViz akan menampilkan berbagai data seperti peta, sensor lidar, dan posisi robot di dalam lingkungan 3D.

### Langkah 4: Memvisualisasikan Data Point Cloud di RViz

Untuk memproses data point cloud yang dihasilkan oleh sensor robot, kita dapat menggunakan PCL dan RViz. Di RViz, kita dapat menambahkan tampilan "PointCloud2" untuk memvisualisasikan data point cloud. Jalankan node yang mengumpulkan data dari sensor LiDAR dengan menggunakan perintah `roslaunch turtlebot_bringup minimal.launch`.

## Langkah 5: Analisis dan Manipulasi Model 3D

Setelah data point cloud dan model 3D ditampilkan di RViz, selanjutnya dapat dilakukan analisis dan manipulasi data tersebut. Misalnya dengan menggunakan PCL untuk melakukan segmentasi atau pemfilteran pada data point cloud, serta menganalisis posisi dan orientasi objek di dalam lingkungan 3D.

## 5. Hasil yang Diperoleh

Setelah mengikuti langkah-langkah di atas, hasil yang diperoleh adalah sebagai berikut:

1. **Simulasi Robot di Gazebo:** Model robot berhasil dimuat dan dapat bergerak di dalam lingkungan 3D yang disimulasikan menggunakan Gazebo.
2. **Visualisasi di RViz:** Data sensor seperti LiDAR dan kamera depth berhasil ditampilkan di RViz dalam bentuk point cloud, memungkinkan visualisasi 3D dari lingkungan robot.
3. **Manipulasi Data Point Cloud:** Data point cloud berhasil diproses dan dianalisis menggunakan alat dari PCL, memberikan wawasan tambahan tentang objek-objek yang ada di lingkungan 3D.
4. **Interaktivitas dengan Model 3D:** Pemrograman model 3D dengan URDF memungkinkan manipulasi objek robot dalam simulasi dengan kontrol yang sangat baik di dalam Gazebo dan RViz.

## 6. Kesimpulan

Pada chapter ini, telah dilakukan analisis cara menggunakan ROS untuk pemodelan 3D dalam aplikasi robotik. ROS, bersama dengan alat seperti Gazebo, RViz, dan PCL, memungkinkan simulasi robot yang sangat realistis dalam lingkungan 3D. Langkah-langkah simulasi dan visualisasi yang dilakukan mengonfirmasi bahwa ROS memberikan platform yang sangat kuat untuk mengembangkan, memvisualisasikan, dan menguji robot dalam ruang tiga dimensi, baik untuk tujuan pengembangan maupun penelitian. Dengan memanfaatkan alat ini, pengembang dapat meningkatkan pemahaman mereka tentang interaksi robot dengan lingkungan serta mengoptimalkan algoritma kontrol.

## Chapter 4. Simulating Robots Using ROS and Gazebo

Simulasi robot menggunakan ROS dan Gazebo adalah salah satu metode yang sangat efektif untuk mengembangkan dan menguji algoritma robotika dalam lingkungan virtual sebelum diterapkan pada perangkat keras nyata. ROS (Robot Operating System) menyediakan framework yang fleksibel untuk pengembangan robot, sedangkan Gazebo adalah simulator 3D yang memungkinkan pengujian robot dalam lingkungan realistis. Chapter ini menjelaskan bagaimana ROS dan Gazebo dapat digunakan untuk mensimulasikan robot, memberikan gambaran mengenai alat yang digunakan, dan bagaimana langkah-langkah simulasi dilakukan untuk memperoleh hasil yang bermanfaat.

### 1. Pengantar tentang ROS dan Gazebo

ROS adalah framework perangkat lunak yang mendukung pengembangan aplikasi robotik secara modular. Dengan menggunakan ROS, pengembang dapat berfokus pada pengembangan algoritma kontrol, komunikasi antar-node, serta integrasi sensor dan aktuator.

Gazebo adalah simulator 3D yang kuat yang menyediakan simulasi fisik dan grafis yang sangat realistis untuk robot. Gazebo dilengkapi dengan berbagai sensor dan alat bantu untuk memodelkan robot, yang membuatnya sangat berguna dalam pengembangan robotika.

ROS dan Gazebo sering digunakan bersama-sama dalam pengembangan robot, karena Gazebo memberikan simulasi robot yang dapat berinteraksi dengan ROS, memungkinkan pengujian sistem robot secara virtual.

### 2. Komponen Utama dalam Simulasi Robot

Untuk melakukan simulasi robot menggunakan ROS dan Gazebo, ada beberapa komponen utama yang perlu dipahami:

- **Gazebo Simulator:** Digunakan untuk mensimulasikan robot dalam lingkungan 3D. Gazebo mendukung fisika, interaksi objek, dan sensor seperti LiDAR, kamera, dan IMU (Inertial Measurement Unit).
- **URDF (Unified Robot Description Format):** Format yang digunakan untuk mendeskripsikan robot dalam bentuk file XML. URDF menggambarkan berbagai komponen robot, seperti sasis, roda, sensor, dan hubungan antar bagian robot.
- **ROS Packages for Gazebo:** Paket ROS yang menghubungkan ROS dengan Gazebo, seperti `gazebo_ros` yang menyediakan interface antara ROS dan simulator Gazebo.
- **RViz:** Alat visualisasi yang digunakan untuk menampilkan data sensor, posisi robot, dan peta yang dihasilkan selama simulasi.

### 3. Persiapan Sistem

Sebelum memulai simulasi, ada beberapa langkah yang perlu dilakukan untuk menyiapkan sistem:

- **Instalasi ROS dan Gazebo:** ROS dapat diinstal dengan perintah `sudo apt-get install ros-<versi-ros>-desktop-full`. Untuk Gazebo, pastikan Gazebo versi yang kompatibel dengan ROS yang digunakan, seperti `sudo apt-get install gazebo11 ros-<versi-ros>-gazebo-ros-pkgs`.
- **Membuat Workspace ROS:** Untuk memulai dengan ROS, buat workspace dengan perintah: `catkin_init_workspace`.
- **Mendapatkan Model Robot:** Pilih model robot yang akan digunakan dalam simulasi. ROS menyediakan berbagai model robot, seperti TurtleBot, yang dapat diunduh dan digunakan dalam simulasi.

## 4. Langkah-langkah Simulasi

Setelah menyiapkan sistem, langkah-langkah berikut dapat diikuti untuk memulai simulasi robot menggunakan ROS dan Gazebo.

### Langkah 1: Membuat Paket ROS

Buatlah paket ROS untuk simulasi robot yang akan dilakukan dimana paket ini berisi konfigurasi robot dan file urdf untuk mendeskripsikan robot dengan menggunakan perintah `catkin_create_pkg robot_simulation gazebo_ros rospy std_msgs`.

### Langkah 2: Menambahkan Model Robot

Kemudian gunakan model robot TurtleBot dalam simulasi. ROS menyediakan model TurtleBot yang siap digunakan dengan menggunakan perintah `sudo apt-get install ros-<versi-ros>-turtlebot`. Paket ini akan menyediakan file URDF untuk mendeskripsikan robot TurtleBot, serta konfigurasi yang diperlukan untuk menjalankan simulasi robot.

### Langkah 3: Menjalankan Gazebo

Setelah paket dan model robot disiapkan, jalankan simulasi robot dalam Gazebo. Untuk memulai Gazebo dengan model robot TurtleBot, jalankan perintah `roslaunch turtlebot_gazebo turtlebot_world.launch`. Perintah ini akan membuka simulator Gazebo dengan model robot TurtleBot yang akan beroperasi dalam lingkungan virtual yang sudah disediakan.

### Langkah 4: Visualisasi dengan RViz

Setelah robot dijalankan dalam Gazebo, gunakan RViz untuk memvisualisasikan data sensor yang diterima oleh robot. Misalnya, untuk menampilkan data dari sensor LiDAR, kamera, dan IMU yang ada di robot, dapat dilakukan dengan perintah `roslaunch turtlebot_rviz_launchers view_navigation.launch`. Dengan perintah ini, RViz akan menampilkan berbagai data yang dipancarkan oleh robot, seperti peta, jalur robot, serta sensor LiDAR.



## Langkah 5: Mengendalikan Robot

Setelah simulasi dimulai, robot dapat dikendalikan dengan menggunakan perintah ROS atau dengan menghubungkan input dari keyboard atau joystick. Untuk mengendalikan robot menggunakan keyboard, kita dapat menjalankan node teleoperasi dengan perintah `roslaunch turtlebot_teleop keyboard_teleop.launch`. Perintah ini berfungsi untuk mengendalikan gerakan robot menggunakan keyboard.

## 5. Hasil yang Diperoleh

Setelah mengikuti langkah-langkah simulasi, hasil yang diperoleh adalah:

1. **Simulasi Robot dalam Gazebo:** Robot berhasil disimulasikan dalam lingkungan 3D dengan menggunakan Gazebo. Robot bergerak dan berinteraksi dengan lingkungan berdasarkan sensor yang dipasang pada robot.
2. **Visualisasi di RViz:** RViz berhasil menampilkan data sensor robot, seperti posisi robot, peta yang dihasilkan, dan data dari sensor LiDAR atau kamera depth.
3. **Kontrol Robot:** Robot berhasil dikendalikan melalui teleoperasi menggunakan keyboard, dan pergerakannya di Gazebo mencerminkan kontrol yang dilakukan.

## 6. Kesimpulan

Pada chapter ini, telah dibahas bagaimana menggunakan ROS dan Gazebo untuk mensimulasikan robot dalam lingkungan virtual 3D. Melalui langkah-langkah yang telah dilakukan, dapat disimpulkan bahwa ROS dan Gazebo adalah kombinasi yang sangat kuat untuk pengembangan dan pengujian robotik. Simulasi robot yang dilakukan memungkinkan pengembangan yang lebih efisien dengan memanfaatkan alat visualisasi dan simulasi yang mendalam. Dengan menggunakan ROS dan Gazebo, pengembang dapat melakukan pengujian robot dalam berbagai skenario tanpa harus mengandalkan perangkat keras fisik, yang menghemat waktu dan biaya.