

Laporan Hasil Analisis Simualsi

Nama : Yusran yasir

NIM : 1103213166

1. Simulasi Getting Started with ROS Programming

Bab 2 dari buku Mastering ROS for Robotics Programming, Third Edition berjudul "Getting Started with ROS Programming" memberikan pembaca pemahaman dasar tentang pemrograman di ROS (Robot Operating System). Bab ini dimulai dengan panduan instalasi ROS di berbagai sistem operasi, terutama Linux, serta pengaturan workspace ROS yang diperlukan untuk pengembangan aplikasi robot. Pembaca diperkenalkan dengan konsep dasar ROS seperti node, topik, layanan, dan parameter server, yang memungkinkan komunikasi antar bagian sistem robot. Selain itu, bab ini juga membahas alat-alat penting seperti roscore, rosrn, roslaunch, rostopic, dan rviz, yang digunakan untuk menjalankan, memonitor, dan memvisualisasikan aplikasi ROS. Selanjutnya, pembaca diberi tutorial untuk menulis aplikasi ROS pertama mereka, seperti membuat publisher dan subscriber untuk komunikasi antar node. Bab ini juga mengajarkan pembaca tentang alur kerja ROS, di mana data dipertukarkan melalui publisher/subscriber atau layanan/client. Meskipun bab ini sangat berguna bagi pemula, beberapa pembaca mungkin menemui tantangan dalam hal konfigurasi dan kompatibilitas perangkat keras. Secara keseluruhan, bab ini memberikan fondasi yang solid untuk mengembangkan aplikasi robotika di ROS dan mempersiapkan pembaca untuk melanjutkan ke topik yang lebih kompleks.

Command yang digunakan :

```
roslaunch mastering_ros_demo_pkg demo_topic_publisher
```

Perintah ini menjalankan node yang berfungsi untuk mempublikasikan pesan (message) pada sebuah topik di ROS. demo_topic_publisher akan mengirimkan data secara berkala ke topik tertentu.

```
roslaunch mastering_ros_demo_pkg demo_topic_subscriber
```

Perintah ini menjalankan node yang berfungsi untuk berlangganan (subscribe) pada sebuah topik di ROS. demo_topic_subscriber akan menerima pesan yang dipublikasikan oleh demo_topic_publisher pada topik yang sama.

```
roslaunch mastering_ros_demo_pkg demo_msg_publisher
```

Perintah ini menjalankan node yang mempublikasikan pesan dengan tipe data tertentu, dalam hal ini adalah pesan yang lebih spesifik daripada sekedar topik biasa. demo_msg_publisher akan mengirimkan pesan dengan format yang telah didefinisikan dalam kode (biasanya dalam file .msg).

```
roslaunch mastering_ros_demo_pkg demo_msg_subscriber
```

Perintah ini menjalankan node yang berlangganan pada pesan dengan tipe data tertentu, sesuai dengan yang dipublikasikan oleh `demo_msg_publisher`. Node ini akan menerima pesan-pesan dengan format tertentu dari publisher yang sesuai.

```
roslaunch mastering_ros_demo_pkg demo_service_server
```

Perintah ini menjalankan node yang berfungsi sebagai server untuk layanan (service). Server ini akan menunggu permintaan dari klien, memprosesnya, dan mengirimkan respons kembali ke klien.

```
roslaunch mastering_ros_demo_pkg demo_service_client
```

Perintah ini menjalankan node yang berfungsi sebagai klien yang mengirimkan permintaan (request) ke server layanan (service). Klien ini akan mengirimkan data ke server dan menerima respons yang dikirimkan oleh `demo_service_server`.

2. Simulasi Working with ROS for 3D Modeling

Bab 3 dari buku *Mastering ROS for Robotics Programming, Third Edition* berjudul "Working with ROS for 3D Modeling" mengajarkan pembaca bagaimana menggunakan ROS untuk menangani pemodelan 3D dalam konteks robotika. Bab ini memberikan pengantar tentang cara bekerja dengan alat dan pustaka ROS untuk membuat, memvisualisasikan, dan mengelola model 3D robot. Pembaca diperkenalkan dengan URDF (Unified Robot Description Format) dan XACRO (XML Macros), yang digunakan untuk menggambarkan model robot dalam format yang dapat diproses oleh ROS. Selain itu, bab ini membahas cara mengonfigurasi sensor, link, dan joint pada model robot untuk mendukung simulasi fisik yang akurat. Bab ini juga mengajarkan cara menggunakan alat visualisasi seperti RViz dan Gazebo untuk memvisualisasikan model 3D dan melakukan simulasi gerakan robot. Fokus utama dari bab ini adalah bagaimana mengintegrasikan model 3D ke dalam alur kerja ROS, memanfaatkan berbagai paket untuk mengatur aspek kinematika dan dinamika robot, serta menguji model dalam simulasi. Pembaca juga diberikan wawasan tentang pentingnya akurasi model dalam pengembangan robot yang efektif, serta cara menangani tantangan yang muncul dalam pembuatan dan pemeliharaan model 3D yang kompleks. Secara keseluruhan, bab ini memberikan panduan yang sangat berguna bagi pengembang yang ingin bekerja dengan model 3D dalam ROS, baik untuk simulasi maupun implementasi robot fisik.

Command yang digunakan :

```
roslaunch mastering_ros_robot_description_pkg view_demo.launch
```

Perintah ini meluncurkan file `view_demo.launch`, yang biasanya dirancang untuk memvisualisasikan demo robot.

```
roslaunch mastering_ros_robot_description_pkg view_arm.launch
```

Perintah ini meluncurkan file `view_arm.launch`, yang fokus pada visualisasi bagian robot manipulator atau lengan robot. File ini sering digunakan untuk melihat model dan pergerakan lengan robot dalam simulasi atau visualisasi di RViz.

```
roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch
```

Perintah ini meluncurkan file `view_mobile_robot.launch`, yang dirancang untuk memvisualisasikan robot bergerak (mobile robot) di RViz. Robot jenis ini biasanya melibatkan roda atau sistem navigasi yang memungkinkan pergerakan di lingkungan tertentu.

3. Simulasi Simulating Robots Using ROS and Gazebo

Bab 4 dari buku *Mastering ROS for Robotics Programming, Third Edition* berjudul "Simulating Robots Using ROS and Gazebo" membahas cara mengintegrasikan ROS dengan Gazebo untuk melakukan simulasi robot yang realistis. Bab ini memberikan pembaca pemahaman mendalam tentang bagaimana menggunakan Gazebo, platform simulasi robot yang populer, bersama dengan ROS untuk membuat simulasi yang mendekati kondisi dunia nyata. Pembaca diajarkan cara mengonfigurasi dan menjalankan simulasi robot dengan menggunakan model 3D yang telah dibuat, serta bagaimana menyetting berbagai sensor dan perangkat keras virtual seperti kamera, LIDAR, dan IMU di dalam simulasi. Bab ini juga mengajarkan cara mengontrol robot dalam simulasi, menggunakan perintah ROS untuk mengirimkan data kecepatan dan posisi, serta berinteraksi dengan Gazebo untuk mengatur parameter simulasi. Selain itu, pembaca diberi instruksi tentang cara memanfaatkan berbagai plugin Gazebo untuk meningkatkan interaksi antara simulasi dan ROS, seperti kontroler untuk gerakan robot dan pembuatan peta. Fokus utama dari bab ini adalah bagaimana memanfaatkan Gazebo untuk menguji algoritma navigasi dan penghindaran rintangan, serta menguji berbagai skenario di dunia maya sebelum mengimplementasikannya pada robot fisik. Dengan panduan praktis dan contoh kode, bab ini memberikan fondasi yang kuat bagi pengembang yang ingin melakukan simulasi robot di ROS dan Gazebo, serta memperkenalkan alat-alat yang dapat digunakan untuk memecahkan masalah dalam simulasi dan pengujian robot.

Command yang digunakan :

```
roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch
```

Perintah ini meluncurkan simulasi robot lengan dengan 7 derajat kebebasan (7-DOF) di simulator Gazebo.

```
rostopic pub /seven_dof_arm/joint4_position_controller/command std_msgs/Float64  
"data: 1.0"
```

Perintah ini mempublikasikan nilai posisi target untuk joint ke-4 pada lengan robot 7-DOF. Joint ini dikontrol oleh pengontrol posisi yang mendengarkan topik `/seven_dof_arm/joint4_position_controller/command`.

```
roslaunch diff_wheeled_robot_control keyboard_teleop.launch
```

Perintah ini meluncurkan kontrol manual untuk robot bergerak (differential wheeled robot) menggunakan input dari keyboard. File `keyboard_teleop.launch` biasanya menginisialisasi node untuk membaca input keyboard dan menerjemahkannya menjadi perintah gerakan.

```
rviz
```

Perintah ini meluncurkan RViz

4. Simulasi Exploring the Advanced Capabilities of ROS MoveIt!

Bab 7 dari buku *Mastering ROS for Robotics Programming, Third Edition* berjudul "Exploring the Advanced Capabilities of ROS MoveIt!" membahas penggunaan MoveIt!, sebuah paket ROS yang kuat untuk perencanaan gerakan dan kontrol manipulasi robot. Bab ini memberikan pembaca pemahaman mendalam tentang kemampuan lanjutan MoveIt!, yang digunakan untuk mengatur pergerakan robot dalam ruang tiga dimensi, termasuk penghindaran tabrakan, perencanaan jalur, dan kontrol lengan robot. Pembaca diperkenalkan dengan fitur-fitur seperti perencanaan gerakan multikonfigurasi, simulasi gerakan secara real-time, dan integrasi dengan alat visualisasi seperti RViz untuk melihat hasil perencanaan gerakan. Selain itu, bab ini mengajarkan cara mengonfigurasi dan menyesuaikan MoveIt! dengan berbagai jenis robot, termasuk lengan manipulasi dan robot mobile, serta cara melakukan kalibrasi dan mengoptimalkan perencanaan jalur. Pembaca juga diberi wawasan tentang cara mengintegrasikan MoveIt! dengan sistem kontrol fisik, sehingga mereka dapat menguji dan mengimplementasikan algoritma gerakan langsung pada robot nyata. Fokus utama bab ini adalah untuk memperkenalkan pembaca pada teknik-teknik canggih dalam perencanaan gerakan dan penghindaran tabrakan, yang esensial untuk pengembangan aplikasi robotika yang kompleks dan efisien. Dengan tutorial praktis dan studi kasus, bab ini memberikan pemahaman yang sangat berguna bagi pengembang yang ingin mengoptimalkan kemampuan perencanaan gerakan dan manipulasi robot menggunakan ROS MoveIt!.

Command yang digunakan :

```
roslaunch seven_dof_arm_config demo.launch
```

Perintah ini digunakan untuk meluncurkan file `demo.launch` yang ada dalam paket `seven_dof_arm_config`.

```
roslaunch seven_dof_arm_test test_random_node
```

Perintah ini menjalankan `test_random_node` yang ada dalam paket `seven_dof_arm_test`.

```
roslaunch seven_dof_arm_test add_collision_object
```

erintah ini menjalankan node `add_collision_object` yang ada dalam paket `seven_dof_arm_test`.

```
roslaunch seven_dof_arm_test pick_place
```

Perintah ini menjalankan node `pick_place` dalam paket `seven_dof_arm_test`, yang biasanya digunakan untuk menguji kemampuan robot dalam tugas mengambil (pick) dan menempatkan (place) objek.