

Nama : Faiz Hibatullah

NIM : 1103210172

Kelas : TK-45-G09

Chapter 2: Memulai Pemrograman ROS

Assalamualaikum Wr Wb saya Faiz Hibatullah dari jurusan Teknik computer. Disini saya akan melakukan tutorial ROS programming yang akan mempelajari dasar-dasar pemrograman ROS melalui pembuatan dan pengelolaan package ROS. Fokus utama adalah memahami bagaimana node ROS berkomunikasi menggunakan topics dan services. Dengan menyelesaikan video ini, kita akan memiliki pemahaman dasar tentang struktur dan fungsi ROS, termasuk penggunaan actionlib untuk menjalankan tugas yang lebih kompleks dengan feedback selama proses eksekusi.

Simulasi langkah-langkah melalui terdiri dari proses mendirikan workspace ROS, menjalankan ROS Master, dan mengimplementasikan node untuk komunikasi data. Jadi untuk Langkah:

- 1. Pastikan Ros sudah terinstall dalam mesin untuk tutorial ini**
- 2. Mengkloning Repository:**

```
git clone https://github.com/PacktPublishing/Mastering-ROS-for-Robotics-Programming-Third-edition
```

Perintah ini mengunduh seluruh kode sumber dari GitHub untuk digunakan dalam pembelajaran ROS.

- 3. Membuat Workspace ROS:**
4. `mkdir -p ~/catkin_ws/src`
5. `cd ~/catkin_ws`
`catkin_make`

Workspace ROS dikelola dengan `catkin_make` untuk membangun semua paket di dalamnya.

- 6. Mengatur Environment:**

```
gedit ~/.bashrc
```

Tambahkan baris berikut di akhir file:

```
source ~/catkin_ws/devel/setup.bash
```

Setelah itu, jalankan:

```
source ~/.bashrc
```

Ini memastikan semua perintah ROS tersedia setiap kali terminal dibuka.

- 7. Menjalankan ROS Master:**

```
roscore
```

ROS Master adalah pusat komunikasi antara semua node.

8. Menguji Publisher dan Subscriber:

- Publisher:

```
roslaunch mastering_ros_demo_pkg demo_topic_publisher
```

Node ini menerbitkan pesan string ke topic /demo_topic.

- Subscriber:

```
roslaunch mastering_ros_demo_pkg demo_topic_subscriber
```

Node ini menerima pesan dari topic /demo_topic dan mencetaknya ke terminal.

9. Menggunakan Custom Message:

- Publisher:

```
roslaunch mastering_ros_demo_pkg demo_msg_publisher
```

Node ini menerbitkan pesan custom yang didefinisikan dalam demo_msg.

- Subscriber:

```
roslaunch mastering_ros_demo_pkg demo_msg_subscriber
```

Node ini membaca pesan custom tersebut.

Chapter 3: Bekerja dengan Pemodelan 3D di ROS

Assalamualaikum Wr Wb. Untuk video ini kita akan memperkenalkan pada pemodelan robot 3D menggunakan URDF dan xacro. Tujuannya adalah untuk membantu kita memahami cara mendeskripsikan struktur robot dan memvisualisasikannya di RViz dan memodifikasi parameter seperti properti fisik dan tabrakan. Dengan mengonversi file xacro ke URDF, kita dapat bereksperimen dengan deskripsi robot yang terparameterisasi, menciptakan model yang fleksibel dan dinamis.

Melalui langkah-langkah ini, kita akan mampu membuat model robot yang parameterisasi, seperti lengan robot 7-DOF atau robot bergerak diferensial, dan memahami bagaimana deskripsi robot diterjemahkan ke dalam simulasi visual.

Langkah-langkah berikut menunjukkan cara memvisualisasikan model robot menggunakan URDF di RViz:

1. Meluncurkan Model Demo:

```
roslaunch mastering_ros_robot_description_pkg view_demo.launch
```

Perintah ini memvisualisasikan robot sederhana dengan URDF di RViz.

2. **Melihat Model Lengan Robot:**

```
roslaunch mastering_ros_robot_description_pkg view_arm.launch
```

Ini memvisualisasikan lengan robot 7-DOF, lengkap dengan joint yang dapat digerakkan.

3. **Melihat Robot Bergerak:**

```
roslaunch mastering_ros_robot_description_pkg  
view_mobile_robot.launch
```

Perintah ini menampilkan robot bergerak diferensial dengan roda yang dapat dikontrol.

Chapter 4: Mensimulasikan Robot Menggunakan ROS dan Gazebo

Assalamualaikum Wr Wb. Video ini akan membawa kita dalam menggunakan Gazebo untuk mensimulasikan robot. Disini kita dapat meluncurkan dunia Gazebo, menambahkan sensor seperti kamera dan LIDAR ke model robot mereka, serta mengontrol robot menggunakan plugin `ros_control`. Kita akan belajar bagaimana meluncurkan simulasi untuk lengan robot 7-DOF dan robot bergerak diferensial. Fokusnya adalah pada kontrol robot melalui plugin `ros_control` dan mempublikasikan data ke topik untuk memindahkan joint tertentu.

langkah-langkah yang akan dilakukan akan memahami bagaimana mengintegrasikan Gazebo dengan ROS untuk simulasi yang interaktif dan realistis, membuka jalan untuk pengembangan aplikasi robot yang lebih kompleks.

1. **Meluncurkan Simulasi Lengan Robot 7-DOF:**

```
roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch
```

Ini memulai simulasi lengan robot di Gazebo dengan kontrol penuh terhadap setiap joint.

2. **Mengontrol Joint Secara Manual:**

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

Perintah ini mengirimkan nilai posisi ke joint ke-4, membuatnya bergerak ke posisi tertentu.

3. **Meluncurkan Robot Diferensial:**

```
roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo.launch
```

Perintah ini akan membuka gazebo dan memulai simulasi robot diferensial.

4. Menggerakkan Robot Diferensial:

```
roslaunch diff_wheeled_robot_control keyboard_teleop.launch
```

Perintah akan membuat node teleop dan memperbolehkan kita untuk menggerakkan robot diferensial.

Chapter 6: Menggunakan ROS MoveIt! dan Navigation Stack

Assalamualaikum Wr Wb. Video ini akan mengajarkan cara menggunakan MoveIt! untuk manipulasi robot dan Navigation Stack untuk navigasi otonom. MoveIt! memungkinkan perencanaan gerakan robot, sedangkan Navigation Stack membantu robot membuat peta dan bernavigasi melalui lingkungan dengan menghindari rintangan.

Di sini akan mempelajari bagaimana mensimulasikan robot bergerak diferensial di Gazebo, membuat peta menggunakan SLAM, mengontrol robot dengan teleop, serta menyimpan dan memanfaatkan peta untuk navigasi otomatis.

1. Meluncurkan Robot Roda Diferensial di Gazebo:

```
roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch
```

Ini memulai simulasi robot bergerak diferensial di dunia Gazebo.

2. Membuat Peta dengan SLAM:

```
roslaunch diff_wheeled_robot_gazebo gmapping.launch
```

Node ini menjalankan SLAM (Simultaneous Localization and Mapping) untuk membuat peta lingkungan.

3. Mengontrol Robot dengan Keyboard:

```
roslaunch diff_wheeled_robot_control keyboard_teleop.launch
```

Perintah ini memungkinkan pengendalian robot secara manual melalui keyboard.

4. Meluncurkan RViz untuk Visualisasi:

```
rviz
```

Gunakan RViz untuk memvisualisasikan peta, posisi robot, dan jalur navigasi.

5. Menyimpan Peta:

```
roslaunch map_server map_saver -f plaza
```

Perintah ini menyimpan peta yang telah dibuat dalam format file untuk digunakan nanti.

