

Nama : Andi Aswan

Nim :1103204095

Dua bab pertama dari buku ini akan memperkenalkan konsep dasar ROS dan sistem manajemen paket ROS untuk melakukan pendekatan terhadap pemrograman ROS. Dalam bab pertama ini , kita akan membahas konsep ROS seperti master ROS, node ROS, node Server parameter ROS, dan pesan dan layanan ROS, sambil membahas apa yang kita butuhkan untuk menginstal ROS dan cara memulai dengan master ROS.

Dalam bab ini, kita akan membahas topik-topik berikut:

- Mengapa kita harus mempelajari ROS?
- Memahami tingkat sistem berkas ROS.
- Memahami tingkat grafik komputasi ROS.
- Tingkat komunitas ROS.

Chapter 1: Introduction to ROS Programming Essentials

Persyaratan teknis

Untuk mengikuti bab ini, satu-satunya hal yang Anda perlukan adalah komputer standar yang menjalankan Ubuntu,20.04 LTS atau distribusi Debian 10 GNU/Linux.

Mengapa kita harus menggunakan ROS?

Robot Operating System (ROS) adalah kerangka kerja fleksibel yang menyediakan berbagai alat dan pustaka untuk menulis perangkat lunak robotik. ROS menawarkan beberapa fitur canggih untuk membantu pengembang dalam tugas-tugas seperti pengiriman pesan, komputasi terdistribusi, penggunaan kembali kode, dan implementasi algoritme canggih untuk aplikasi robotik. Proyek ROS dimulai pada tahun 2007 oleh Morgan Quigley dan pengembangannya dilanjutkan di Willow Garage, sebuah laboratorium untuk mengembangkan perangkat keras dan perangkat lunak sumber terbuka untuk robot. Tujuan dari ROS adalah untuk menetapkan cara standar untuk memprogram robot sambil menawarkan perangkat lunak siap pakai yang dapat dengan mudah diintegrasikan dengan aplikasi robotik khusus. Ada banyak alasan untuk memilih ROS sebagai kerangka kerja pemrograman, dan beberapa di antaranya adalah sebagai berikut:

- **Kemampuan kelas atas:** ROS hadir dengan fungsi yang siap digunakan. Sebagai contoh, paket **Simultaneous Localization and Mapping (SLAM)** dan **Adaptive Monte Carlo Localization (AMCL)** di ROS dapat digunakan untuk memiliki otonom

navigasi di robot seluler, sedangkan paket MoveIt dapat digunakan untuk gerakan perencanaan untuk manipulator robot. Kemampuan ini dapat langsung digunakan dalam perangkat lunak robot tanpa kerumitan. Dalam beberapa kasus, paket-paket ini cukup untuk memiliki tugas robotika inti pada platform yang berbeda. Juga, kemampuan ini sangat tinggi dapat dikonfigurasi; kita dapat menyempurnakan masing-masing menggunakan berbagai parameter.

- **Banyak sekali alat:** Ekosistem ROS dikemas dengan banyak sekali alat untuk melakukan debugging, memvisualisasikan, dan melakukan simulasi. Alat-alat tersebut, seperti **rqt_gui**, **RViz**, dan **Gazebo**, adalah beberapa alat open source terkuat untuk debugging, visualisasi, dan simulasi. Kerangka kerja perangkat lunak yang memiliki banyak alat ini sangat jarang.

- **Dukungan untuk sensor dan aktuator kelas atas:** ROS memungkinkan kita untuk menggunakan perangkat yang berbeda driver dan paket antarmuka berbagai sensor dan aktuator dalam robotika. Seperti itu sensor kelas atas termasuk LIDAR 3D, pemindai laser, sensor kedalaman, aktuator, dan banyak lagi. Kami dapat menghubungkan komponen-komponen ini dengan ROS tanpa kerumitan.

- **Pengoperasian antar platform:** Middleware pengirim pesan ROS memungkinkan komunikasi antara program yang berbeda. Di ROS, middleware ini dikenal sebagai node. Node ini dapat diprogram dalam bahasa apa pun yang memiliki klien ROS perpustakaan. Kita dapat menulis node high-level di C++ atau C dan node lain di Python atau Java.

- **Modularitas:** Salah satu masalah yang dapat terjadi di sebagian besar robot mandiri aplikasi adalah bahwa jika salah satu utas dari kode utama macet, seluruh robot aplikasi dapat berhenti. Di ROS, situasinya berbeda; kami menulis yang berbeda-beda untuk setiap proses, dan jika satu node macet, sistem masih dapat bekerja.

Chapter 2: Getting Started with ROS Programming

Membangun Paket ROS:

➤ Membuat Ruang Kerja Catkin

- 1) Buat direktori *catkin_ws/src*.
- 2) Jalankan *catkin_init_workspace*.
- 3) Jalankan *source /opt/ros/noetic/setup.bash*.
- 4) Jalankan *cd ~/catkin_ws*.

➤ Membuat Paket

- 1) Jalankan *catkin_create_pkg package_name [dependency1] [dependency2]* di dalam folder *~/catkin_ws/src*.
- 2) Contoh: *catkin_create_pkg mastering_ros_demo_pkg roscpp std_msgs actionlib actionlib_msgs*.

➤ Membuat Node

- 1) Buat file kode sumber, misal *demo_topic_publisher.cpp*.
- 2) Edit file *CMakeLists.txt* untuk mengkompilasi kode.
- 3) Jalankan *catkin_make* untuk membangun paket.

➤ Menjalankan Node

- 1) Jalankan *roscore*.
- 2) Jalankan *roslaunch mastering_ros_demo_pkg demo_topic_publisher* di terminal terpisah.
- 3) Jalankan *roslaunch mastering_ros_demo_pkg demo_topic_subscriber* di terminal terpisah.

➤ **Membuat Server dan Klien Aksi**

- 1) Buat file *demo_action_server.cpp* dan *demo_action_client.cpp*.
- 2) Edit file *package.xml* dan *CMakeLists.txt*.
- 3) Jalankan *catkin_make*.
- 4) Jalankan *roscore*.
- 5) Jalankan *roslaunch mastering_ros_demo_pkg demo_action_server*.
- 6) Jalankan *roslaunch mastering_ros_demo_pkg demo_action_client 10 1*.

➤ **Membuat File Peluncuran**

- 1) Buat file *demo_topic.launch* di dalam folder
~/catkin_ws/src/mastering_ros_demo_pkg/launch.
- 2) Tambahkan node *demo_topic_publisher* dan *demo_topic_subscriber* ke dalam file.
- 3) Jalankan *roslaunch mastering_ros_demo_pkg demo_topic.launch*.
- 4) Periksa daftar node dengan *rostopic list*.
- 5) Debug node dengan *rqt_console*.

Chapter 3: Working with ROS for 3D Modeling

Membuat Paket ROS untuk Deskripsi Robot:

➤ Membuat Paket ROS

- 1) Sebelum membuat file URDF untuk robot, mari buat terlebih dahulu paket ROS di dalam catkin. Jalankan perintah berikut:

```
catkin_create_pkg mastering_ros_robot_description_pkg roscpp tf geometry_msgs urdf rviz xacro
```

- 2) Paket ini bergantung pada paket urdf dan xacro. Jika paket ini belum terinstal di sistem, maka dapat menginstalnya menggunakan manajer paket:

```
sudo apt-get install ros-noetic-urdf  
sudo apt-get install ros-noetic-xacro
```

➤ Menjelaskan file URDF

- 1) Simpan kode URDF sebelumnya sebagai *pan_tilt.urdf* dan periksa apakah file urdf tersebut mengandung kesalahan saat menggunakan perintah berikut:

```
check_urdf pan_tilt.urdf
```

- 2) Untuk menggunakan perintah ini, paket liburdfdom-tools harus diinstal terlebih dahulu. maka dapat menginstalnya menggunakan perintah berikut:

```
sudo apt-get install liburdfdom-tools
```

➤ Melihat Struktur Robot

- 1) Jika ingin melihat struktur tautan dan sambungan robot secara grafis, maka dapat menggunakan alat baris perintah yang disebut *urdf_to_graphviz*:

```
urdf_to_graphviz pan_tilt.urdf
```

- 2) Perintah ini akan menghasilkan dua file: *pan_tilt.gv* dan *pan_tilt.pdf*. anda dapat melihat struktur robot ini dengan menggunakan perintah ini:

```
evince pan_tilt.pdf
```

➤ Melihat Model Robot 3D di Rviz

Meluncurkan model robot di Rviz menggunakan perintah berikut:

```
roslaunch mastering_ros_robot_description_pkg view_demo.launch
```

➤ **Melihat Lengan Tujuh-DOF di Rviz**

Buat file peluncuran berikut di dalam folder peluncuran, dan bangun paket menggunakan perintah *catkin_make*.

```
<launch>
```

```
<param name="robot_description" command="(findmastering ros robot description
pkg)/urdf/arm.urdf"/><nodename="robot state publisher"pkg="robot state
publisher"type="robot state
publisher"/><0><nodename="rviz"pkg="rviz"type="rviz"args="-d<1>(find
mastering_ros_robot_description_pkg)/launch/view_arm.rviz"/>
</launch>
```

➤ **Luncurkan URDF menggunakan perintah berikut:**

roslaunch mastering_ros_robot_description_pkg view_arm.launch

➤ **Melihat Robot Seluler di Rviz**

Jika ingin melihat robot seluler bergerak menggunakan perintah berikut:

roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch

Chapter 4: Simulating Robots Using ROS and Gazebo

Simulasi Lengan Robot dengan Gazebo dan ROS:

➤ Menginstal Paket

Sebelum memulai, instal paket-paket berikut:

```
sudo apt-get install ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-msgs ros-noetic-gazebo-plugins ros-noetic-gazebo-roscontro
```

➤ Membuat Model Simulasi

Perbarui deskripsi robot yang sudah ada dengan menambahkan parameter simulasi.

Buat paket untuk simulasi dengan perintah:

```
catkin_create_pkg seven_dof_arm_gazebo gazebo_msgs gazebo_plugins gazebo_ros gazebo_ros_control mastering_ros_robot_description_pkg
```

➤ Simulasi Lengan Robot dengan Xtion Pro

1) Luncurkan simulasi lengkap dengan perintah:

```
roslaunch seven_dof_arm_gazebo seven_dof_arm_with_rgbd_world.launch
```

2) Lihat data point cloud dari sensor di RViz dengan perintah:

```
rosviz -f /rgbd_camera_optical_frame
```

➤ Meluncurkan Pengontrol ROS dengan Gazebo

Luncurkan file peluncuran untuk pengontrol ROS:

```
roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch
```

➤ Memindahkan Sendi Robot

1) Publikasikan nilai sendi yang diinginkan ke topik perintah pengontrol posisi sendi:

```
rostopic pub /seven_dof_arm/joint4_position_controller/command  
std_msgs/Float64 1.0
```

2) Lihat keadaan sendi robot:

```
rostopic echo /seven_dof_arm/joint_states
```

➤ **Simulasi Robot Beroda Diferensial di Gazebo**

Luncurkan simulasi robot beroda diferensial:

roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo.launch

➤ **Menambahkan Node Teleop ROS**

Node teleop ROS menerbitkan perintah ROS Twist dengan mengambil input keyboard.

➤ **Menginstal Paket**

sudo apt-get install ros-noetic-joy

➤ **Menggerakkan Robot**

1) Luncurkan Gazebo:

roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch

2) Mulai node teleop:

roslaunch diff_wheeled_robot_control keyboard_teleop.launch

3) Mulai RViz:

roslaunch rviz

4) Gunakan keyboard untuk menggerakkan robot.

Chapter 5: Simulating Robots Using ROS, CoppeliaSim, and Webots

Pengaturan CoppeliaSim dengan ROS

➤ Menginstal CoppeliaSim

- 1) Unduh CoppeliaSim dari <http://www.coppeliarobotics.com/downloads.html>.
- 2) Ekstrak file dan ubah nama folder menjadi *CoppeliaSim*.
- 3) Atur variabel lingkungan *COPPELIASIM_ROOT* ke folder CoppeliaSim.
- 4) Jalankan *roscore* sebelum membuka CoppeliaSim.
- 5) Buka CoppeliaSim dengan *./coppeliaSim.sh*.

➤ Simulasi Lengan Robot di CoppeliaSim

- 1) Konversi model xacro lengan ke file URDF.
- 2) Simpan file URDF di folder *urdf* pada paket *csim_demo_pkg*.
- 3) Luncurkan simulasi dengan *roslaunch csim_demo_pkg seven_dof_arm.launch*.

Pengaturan Webots dengan ROS

➤ Menginstal Webots

1. Autentikasi repositori Cyberbotics: *wget -qO- https://cyberbotics.com/Cyberbotics.asc | sudo apt-key add -*.
2. Tambahkan repositori Cyberbotics: *sudo apt-add-repository 'deb https://cyberbotics.com/debian/ binary-amd64/'* *sudo apt-get update*.
3. Instal Webots: *sudo apt-get install webots*

➤ Memulai Webots

1. Jalankan Webots dengan *\$ webots*.

➤ Simulasi Lengan Robot di Webots

1. Instal paket *webots_ros*: *sudo apt-get install ros-noetic-webots-ros*.
2. Buat paket *webots_demo_pkg* dengan dependensi *webots_ros*: *catkin_create_pkg webots_demo_pkg roscpp webots_ros geometry_msgs*.

➤ **Menulis Node Teleop dengan Webots**

1. Buat file *teleop_node.cpp* di paket *webots_demo_pkg*.
2. Implementasikan node ROS yang menerima pesan *geometry_msgs::Twist* dan mengontrol kecepatan roda robot.
3. Bangun paket dengan *catkin_make*.
4. Jalankan node teleop dengan *roslaunch webots_demo_pkg teleop_node*.