AHMAD LUTHFI MUHAJIR // 1103200148

Berikut adalah laporan penjelasan dari **Chapter 1 hingga Chapter 4** pada buku *Mastering ROS for Robotics Programming (Third Edition)*

Laporan Penjelasan: Mastering ROS for Robotics Programming (Third Edition) - Chapter 1 to Chapter 4

Chapter 1: Getting Started with ROS

Tujuan Bab:

Bab pertama bertujuan untuk memberikan pemahaman dasar tentang ROS, menginstal ROS pada sistem yang didukung, serta menjalankan aplikasi pertama di ROS.

Penjelasan:

Apa itu ROS?

Robot Operating System (ROS) adalah kerangka kerja perangkat lunak yang digunakan untuk mengembangkan aplikasi robotika. ROS menyediakan berbagai pustaka dan alat untuk pengolahan data, kontrol perangkat keras, serta antarmuka perangkat lunak. ROS memiliki beberapa komponen utama:

- **Node**: Proses atau program yang menjalankan fungsi tertentu dalam sistem.
- **Topic**: Media komunikasi untuk pengiriman pesan antara node.
- **Service**: Sistem komunikasi sinkron antara node (request-response).
- **Action**: Digunakan untuk tugas yang memerlukan waktu lebih lama.

Instalasi ROS

1. **ROS 1 (Noetic) pada Ubuntu 20.04**: Untuk menginstal ROS Noetic pada Ubuntu 20.04, Anda perlu menambahkan repositori ROS, memperbarui daftar paket, dan menginstal paket ROS Desktop-Full.
2. **Memulai Node ROS**:
    - Jalankan roscore di terminal untuk memulai ROS Master.
    - Jalankan node turtlesim dengan perintah rosrun turtlesim turtlesim_node.
    - Gunakan rosrun turtlesim turtle_teleop_key untuk mengontrol turtle menggunakan keyboard.

Chapter 2: Introduction to ROS and Workspace Management

Tujuan Bab:

Bab kedua bertujuan untuk memperkenalkan komponen inti ROS serta cara membuat dan mengelola workspace dan paket.

Penjelasan:

Komponen Utama ROS:

- **Master ROS**: Bertanggung jawab untuk mengatur komunikasi antar node.
- **Parameter Server**: Tempat menyimpan konfigurasi yang bisa diakses oleh semua node.
- **Node**: Unit kerja yang berinteraksi dengan node lain menggunakan topic, service, atau action.
- **Messages**: Format data yang digunakan untuk berkomunikasi antar node.

Manajemen Workspace dan Paket:

1. **Membuat Workspace**: Workspace adalah direktori tempat kita bekerja dengan ROS. Workspace bisa dibuat dengan perintah:
2. mkdir -p ~/catkin_ws/src
3. cd ~/catkin_ws
4. catkin_make
5. **Membuat Paket**: Paket adalah unit penyimpanan kode yang berisi program atau aplikasi. Paket dapat dibuat dengan perintah:
6. cd ~/catkin_ws/src
7. catkin_create_pkg beginner_tutorials std_msgs rospy roscpp

Mengkompilasi Workspace:

Workspace dapat dikompilasi dengan perintah:

cd ~/catkin_ws

catkin_make

Chapter 3: ROS Basics, Communication, and Computation Graph

Tujuan Bab:

Bab ini bertujuan untuk memperkenalkan cara kerja komunikasi ROS menggunakan **Topic**, **Service**, dan **Action**.

Penjelasan:

Komunikasi dengan Topic:

- **Publisher**: Program yang mengirimkan pesan.
- **Subscriber**: Program yang menerima pesan.
- **Contoh Publisher (Python)**:
- import rospy
- from std_msgs.msg import String
-
- rospy.init_node('talker')
- pub = rospy.Publisher('chatter', String, queue_size=10)
- rate = rospy.Rate(1) # 1 Hz
- while not rospy.is_shutdown():
- pub.publish("Hello ROS")
- rate.sleep()
- **Contoh Subscriber (Python)**:
- import rospy
- from std_msgs.msg import String
-
- def callback(msg):
- rospy.loginfo(f"I heard {msg.data}")
-
- rospy.init_node('listener')
- rospy.Subscriber('chatter', String, callback)
- rospy.spin()

Menggunakan Service:

- **Definisi Service**: Service memungkinkan komunikasi sinkron antara node.
- **Contoh Pembuatan Service**:
    - Definisikan file service (AddTwoInts.srv) untuk penjumlahan dua angka.

- Buat server dan client untuk menjalankan service yang menerima permintaan dan memberikan respons.

Debugging Tools:

- Melihat daftar node: rosnode list
- Melihat daftar topic: rostopic list

Chapter 4: Working with 3D Robot Models in ROS

Tujuan Bab:

Bab ini bertujuan untuk memperkenalkan cara bekerja dengan model robot 3D menggunakan **URDF**, **RViz**, dan **Gazebo**.

Penjelasan:

URDF (Unified Robot Description Format):

URDF adalah format XML yang digunakan untuk mendeskripsikan model fisik robot. Anda dapat mendefinisikan **link** (bagian dari robot) dan **joint** (hubungan antara dua link).

Contoh file URDF:

```
<robot name="robot_name">
  <link name="base_link"/>
  <joint name="joint1" type="revolute">
    <parent link="base_link"/>
    <child link="link1"/>
    <origin xyz="0 0 0.1" rpy="0 0 0"/>
  </joint>
</robot>
```

Visualisasi di RViz:

RViz adalah alat visualisasi ROS yang digunakan untuk memvisualisasikan data sensor dan model robot. Anda dapat memuat model robot URDF ke dalam RViz untuk melihat struktur 3D robot.

Menjalankan Visualisasi dengan RViz:

```
roslaunch urdf_tutorial display.launch model:=your_model.urdf
```

Simulasi dengan Gazebo:

Gazebo adalah simulator yang memungkinkan Anda untuk mensimulasikan robot dalam lingkungan 3D. Untuk menjalankan Gazebo, Anda dapat menggunakan:

roslaunch gazebo_ros empty_world.launch

Kesimpulan

Bab 1 hingga 4 memberikan pemahaman yang sangat penting dalam mengembangkan aplikasi robotika menggunakan ROS. Di bab pertama, kita mempelajari dasar-dasar ROS dan cara menginstalnya. Bab kedua membahas pembuatan dan pengelolaan workspace serta paket. Bab ketiga menjelaskan dasar komunikasi di ROS, termasuk topic dan service, sedangkan bab keempat mengajarkan tentang penggunaan URDF, RViz, dan Gazebo untuk bekerja dengan model robot 3D.

Jika Anda ingin melanjutkan pembelajaran ke bab selanjutnya atau membutuhkan penjelasan lebih lanjut, Anda dapat melanjutkan mengikuti langkah-langkah praktis di buku ini atau bertanya lebih lanjut.