Naskah Video Chapter 2

Halo, Selamat datang rekan rekan semua, Perkenalkan nama saya Kavilla Zota Qurzian. Sebelumnya saya memohon maaf karena saya tidak dapat memberikan visualisasi langsung pada ubuntu karena terdapat masalah teknis pada laptop saya, jadi saya hanya dapat membuat video tutorial dari handphone saja. Namun saya akan berusaha memberikan penjelasan tentang ROS Berdasarkan buku "Mastering ROS" dengan visualisasi AI. Mohon maaf untuk ketidaknyamanannya.

1. Setelah memahami dasar-dasar ROS pada chapter sebelumnya, sekarang saatnya kita terjun langsung ke praktik pemrograman menggunakan ROS. Di chapter ini, kita akan mempelajari cara membuat package, mendefinisikan pesan dan layanan khusus, menjalankan node, hingga meluncurkan beberapa node sekaligus menggunakan file peluncuran. Mari kita mulai perjalanan ini dengan membuat workspace ROS sebagai langkah pertama.

2. Dalam ROS, package adalah unit terkecil yang mengelompokkan semua file yang dibutuhkan untuk menjalankan sebuah proyek. Sebuah package dapat berisi folder src untuk kode sumber, msg untuk definisi pesan, srv untuk definisi layanan, launch untuk file peluncuran, dan file penting lainnya seperti CMakeLists.txt dan package.xml. Struktur ini memungkinkan pengembang untuk mengelola proyek secara modular, memastikan bahwa setiap bagian memiliki tempatnya masing-masing.

3. Sebelum membuat package, kita perlu menyiapkan workspace terlebih dahulu. Workspace adalah lingkungan kerja tempat kita menyimpan dan mengatur berbagai package. Prosesnya dimulai dengan membuat direktori workspace, menginisialisasinya sebagai workspace ROS, dan membangunnya menggunakan catkin_make. Berikut adalah perintah-perintahnya

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
catkin_init_workspace
cd ~/catkin_ws
catkin_make
```

Setelah itu, kita perlu memastikan workspace ini dikenali oleh sistem dengan menjalankan perintah:

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

Perintah ini menghubungkan workspace kita dengan lingkungan ROS.

4. Dalam ROS, semua proses komputasi dilakukan melalui node. Node adalah program independen yang saling berkomunikasi melalui topik, layanan, atau parameter. Misalnya, pada robot sederhana, kita bisa memiliki node untuk mengolah data dari kamera, node lain untuk menghitung posisi robot, dan node lainnya untuk mengendalikan motor. Struktur ini membuat sistem lebih fleksibel, karena jika satu node berhenti bekerja, node lain tetap bisa berjalan."
   Visual: Diagram node yang saling terhubung melalui topik. Sebuah node ditunjukkan sebagai lingkaran, dan topik sebagai persegi Panjang

5. Selain menggunakan pesan standar yang sudah ada di ROS, kita juga bisa mendefinisikan pesan dan layanan custom. Pesan digunakan untuk mengirimkan data antara node melalui topik. File .msg mendefinisikan struktur data tersebut, misalnya:

   ```
   int32 number
   string name
   float32 speed
   ```

   Sedangkan layanan memungkinkan komunikasi request-response antara node. Format file .srv adalah:

   ```
   # Request
   int32 input
   ---
   # Response
   int32 output
   ```

   Dengan mendefinisikan pesan dan layanan sendiri, kita dapat membuat komunikasi antar-node yang lebih spesifik sesuai kebutuhan proyek kita.

6. Saat bekerja dengan proyek besar yang melibatkan banyak node, akan merepotkan jika kita harus menjalankan setiap node secara manual. Untuk itu, ROS menyediakan file peluncuran. File ini berformat XML dan memungkinkan kita menjalankan beberapa node sekaligus. Contohnya:

   ```
   <launch>
     <node pkg=\"my_package\" type=\"camera_node\" name=\"camera\"/>
     <node pkg=\"my_package\" type=\"image_processor_node\" name=\"image_processor\"/>
   </launch>
   ```

   File ini meluncurkan dua node sekaligus: satu untuk mengambil gambar dari kamera, dan satu lagi untuk memproses gambar tersebut. Dengan file peluncuran, kita dapat

menghemat waktu dan mengelola proyek dengan lebih efisien."

7. Dalam chapter ini, kita telah belajar tentang langkah-langkah awal dalam pemrograman ROS, mulai dari membuat workspace, menyiapkan package, menjalankan node, hingga menggunakan file peluncuran. Dengan memahami konsep-konsep ini, Anda sudah memiliki dasar yang kuat untuk membangun aplikasi robotika yang kompleks. Teruslah bereksplorasi, dan jangan ragu untuk mencoba berbagai fitur ROS lainnya!"
   Visual: Animasi sistem ROS yang aktif, dengan node-node saling terhubung dan data yang mengalir.