Chapter 2
Judul Video:

Mengenal Komunikasi ROS: Publisher, Subscriber, dan Service Menggunakan mastering_ros_demo_pkg

Pembukaan (0:00 - 0:20)

Visual: Tampilan folder mastering_ros_demo_pkg di file manager.

Voiceover: "Halo semua! Kali ini, kita akan menjalankan fitur komunikasi ROS menggunakan paket **mastering_ros_demo_pkg**. Kita akan eksplorasi publisher, subscriber, serta service bawaan paket ini. Yuk kita mulai!"

Bagian 1: Publisher dan Subscriber untuk Message (0:20 - 2:00)

1. Menjalankan Publisher

Visual: Terminal terbuka, menjalankan perintah berikut:

bash

Salin kode

rosrun mastering_ros_demo_pkg demo_msg_publisher

Voiceover: "Pertama, kita jalankan node publisher bawaan untuk mengirim pesan. Gunakan perintah ini di terminal:"

Visual: Output terminal menunjukkan pesan yang diterbitkan.

Voiceover: "Node ini akan menerbitkan pesan secara berkala ke topik tertentu. Untuk melihat topik yang digunakan, jalankan perintah rostopic list di terminal lain."

2. Menjalankan Subscriber

Visual: Terminal kedua, menjalankan perintah berikut:

bash

Salin kode

rosrun mastering_ros_demo_pkg demo_msg_subscriber

Voiceover: "Selanjutnya, kita jalankan node subscriber untuk menerima pesan yang diterbitkan oleh publisher tadi. Gunakan perintah ini:"

Visual: Output terminal menunjukkan pesan diterima oleh subscriber.

Voiceover: "Subscriber ini akan mendengarkan topik yang sama dan menampilkan isi pesan di terminal."

Bagian 2: Publisher dan Subscriber untuk Topik (2:00 - 3:00)

1. Menjalankan Topic Publisher

Visual: Terminal baru, menjalankan perintah berikut:

bash

Salin kode

rosrun mastering_ros_demo_pkg demo_topic_publisher

Voiceover: "Kita juga bisa menerbitkan data ke topik tertentu menggunakan node publisher untuk topik. Gunakan perintah berikut di terminal:"

Visual: Output terminal menunjukkan data yang diterbitkan ke topik.

2. Menjalankan Topic Subscriber

Visual: Terminal lain, menjalankan perintah berikut:

bash

Salin kode

rosrun mastering_ros_demo_pkg demo_topic_subscriber

Voiceover: "Untuk membaca data dari topik tersebut, kita jalankan node subscriber untuk topik. Gunakan perintah ini:"

Visual: Output terminal menunjukkan data diterima oleh subscriber.

Voiceover: "Dengan ini, kita telah melihat bagaimana komunikasi antar-node dilakukan menggunakan topik di ROS."

Bagian 3: Service Server dan Client (3:00 - 4:30)

1. Menjalankan Service Server

Visual: Terminal baru, menjalankan perintah berikut:

bash

Salin kode

rosrun mastering_ros_demo_pkg demo_service_server

Voiceover: "Selanjutnya, kita coba fitur service. Pertama, jalankan service server dengan perintah ini:"

Visual: Terminal menunjukkan bahwa service aktif.

Voiceover: "Service server ini menunggu permintaan dari client untuk memberikan respons."

2. Menjalankan Service Client

Visual: Terminal kedua, menjalankan perintah berikut:

bash

Salin kode

rosrun mastering_ros_demo_pkg demo_service_client

Voiceover: "Setelah server aktif, kita jalankan client untuk mengirim permintaan ke service server. Gunakan perintah ini:"

Visual: Output terminal menunjukkan respons dari server.

Voiceover: "Service client mengirimkan data ke server, dan server merespons sesuai dengan logika yang telah didefinisikan."

Penutup (4:30 - 5:00)

Visual: Terminal menunjukkan semua fitur berhasil dijalankan.

Voiceover: "Demikian tutorial menjalankan publisher, subscriber, serta service bawaan dari paket **mastering_ros_demo_pkg**. Semoga membantu eksplorasi ROS Anda. Jangan lupa like dan subscribe untuk lebih banyak video tutorial! Sampai jumpa!"

Chapter 5

Presenter: "Selamat datang di video tutorial kami! Hari ini, kita akan belajar bagaimana menggunakan *Robot Operating System* atau ROS untuk mengendalikan robot arm di CoppeliaSim. Tutorial ini sangat cocok untuk Anda yang ingin memahami konsep dasar integrasi ROS dengan simulator robot."

[Cut to: Close-up layar komputer menampilkan terminal Linux]

Presenter (Voiceover): "Langkah pertama, kita perlu menjalankan *roscore*. Roscore adalah pusat komunikasi di ROS yang mengelola semua node dan topik yang berjalan."

bash

roscore

Presenter (Voiceover): "Jalankan perintah ini di terminal, dan pastikan tidak ada error."

[Transition: Terminal menampilkan perubahan direktori]

Presenter: "Selanjutnya, kita masuk ke direktori tempat CoppeliaSim berada. Dalam contoh ini, direktori ada di folder Downloads."

bash

cd ~/Downloads/CoppeliaSim_Edu_V4_8_0_rev0_Ubuntu20_04

[Cut to: Terminal membuka CoppeliaSim]

Presenter: "Setelah masuk, kita jalankan file *coppeliaSim.sh* untuk membuka simulasi."

bash

./coppeliaSim.sh

[Cut to: Simulasi CoppeliaSim terbuka, menampilkan robot arm]

Presenter (Voiceover): "Ketika CoppeliaSim sudah berjalan, pastikan Anda telah mengatur robot arm di dalam scene. Dalam tutorial ini, kita akan menggunakan *seven_dof_arm*."

[Cut to: Terminal dengan perintah ROS]

Presenter: "Untuk memvisualisasikan data gambar dari robot, kita dapat menggunakan *rqt_image_view*. Ini akan membantu kita melihat apa yang ditangkap oleh sensor kamera pada robot, jika ada."

bash

rqt_image_view

[Cut to: Terminal dengan perintah rostopic]

Presenter: "Kini saatnya mengendalikan salah satu *joint* pada robot arm. Kita bisa menggunakan perintah berikut untuk menggerakkan *elbow_pitch joint*. Misalnya, kita mengirimkan nilai 1.0 sebagai target posisi."

bash

rostopic pub /csim_demo/seven_dof_arm/elbow_pitch/cmd std_msgs/Float32 "data: 1.0"

Presenter (Voiceover): "Perintah ini mengirimkan pesan ke topik *cmd* milik *elbow_pitch* dengan tipe data Float32."

[Cut to: Terminal membaca status joint]

Presenter: "Untuk memastikan bahwa pergerakan berhasil, kita dapat mengecek status dari *elbow_pitch joint* menggunakan perintah berikut:"

bash

rostopic echo /csim_demo/seven_dof_arm/elbow_pitch/state

Presenter (Voiceover): "Output di terminal akan menunjukkan posisi terkini dari *elbow_pitch joint*. Ini berguna untuk memverifikasi apakah perintah yang kita kirimkan sudah diterima oleh simulator."

[Cut to: Presenter kembali ke layar penuh]

Presenter: "Dengan langkah-langkah ini, Anda sekarang dapat mengontrol robot arm menggunakan ROS dan CoppeliaSim! Eksperimen ini adalah dasar untuk proyek yang lebih kompleks, seperti kontrol robot otomatis atau integrasi dengan sensor lainnya."

[Closing Scene: Presenter tersenyum sambil melambaikan tangan]

Presenter: "Terima kasih telah menonton tutorial ini. Jangan lupa untuk like, share, dan subscribe agar tidak ketinggalan video tutorial berikutnya. Sampai jumpa!"

Chapter 6

Naskah Video: Demonstrasi Seven DOF Arm menggunakan ROS

[Opening Scene: Kamera fokus pada desktop yang menampilkan terminal Linux.]

Narator: "Selamat datang di video tutorial singkat demonstrasi Seven Degree of Freedom Arm menggunakan Robot Operating System atau ROS. Dalam video ini, kita akan menjalankan dua perintah utama untuk memvisualisasikan dan mengontrol lengan robot ini."

[Cut Scene: Tampilan desktop dengan terminal terbuka. Perintah pertama diketik.]

Narator: "Langkah pertama, kita akan menjalankan perintah berikut di terminal:"

[Close-Up: Terminal menampilkan perintah.]

roslaunch seven_dof_arm_config demo.launch

Narator: "Perintah ini akan membuka antarmuka MoveIt! untuk melakukan konfigurasi dan simulasi lengan robot. Pastikan ROS sudah terinstal dan semua dependensi proyek telah terpenuhi. Tekan enter untuk menjalankan."

[Scene: Antarmuka MoveIt! terbuka di layar.]

Narator: "Setelah menjalankan perintah tersebut, Anda akan melihat antarmuka MoveIt! yang menunjukkan model lengan robot. Antarmuka ini memungkinkan kita untuk menguji pergerakan lengan robot secara visual."

[Cut Scene: Kembali ke terminal untuk menjalankan perintah kedua.]

Narator: "Selanjutnya, kita akan menjalankan perintah kedua di terminal."

[Close-Up: Terminal menampilkan perintah kedua.]

roslaunch seven_dof_arm_gazebo seven_dof_arm_bringup_moveit.launch

Narator: "Perintah ini akan membawa simulasi ke dalam lingkungan Gazebo sekaligus mengintegrasikan kontrol MoveIt! untuk lengan robot. Tekan enter untuk memulai."

[Scene: Gazebo terbuka di layar dengan model lengan robot terlihat dalam simulasi.]

Narator: "Gazebo akan menampilkan lingkungan simulasi tiga dimensi. Di sini, kita bisa memantau bagaimana lengan robot bergerak sesuai dengan perintah yang diberikan melalui MoveIt!."

[Cut Scene: Fokus pada pergerakan lengan robot di Gazebo.]

Narator: "Dengan menggabungkan antarmuka MoveIt! dan simulasi Gazebo, kita dapat memvalidasi pergerakan lengan robot secara virtual sebelum diimplementasikan pada perangkat keras nyata."

[Closing Scene: Kamera fokus kembali ke desktop, menunjukkan kedua jendela simulasi dan antarmuka yang berjalan bersamaan.]

Narator: "Itulah tutorial singkat tentang cara menjalankan simulasi Seven DOF Arm menggunakan ROS. Video ini membantu Anda memahami bagaimana konfigurasi, visualisasi, dan simulasi robot dapat dilakukan secara efisien. Sampai jumpa di video tutorial berikutnya!"

[Closing: Layar menampilkan teks "Terima Kasih Sudah Menonton" dengan logo ROS dan Seven DOF Arm.]

Chapter 7

Naskah Video Tutorial: "Menggunakan Robot Lengan dengan ROS"

[Opening Scene]

- Narasi: "Halo semuanya! Di video ini, kita akan belajar cara menggunakan ROS untuk mengoperasikan robot lengan dengan 7 DOF atau Degree of Freedom. Kita akan meliputi proses menjalankan simulasi, menambahkan objek kolisi, hingga melakukan tugas pick and place!"

[Scene 1: Menjalankan Simulasi Robot Lengan]

- Visual: Editor terminal terbuka, menampilkan perintah.
- Narasi: "Langkah pertama adalah menjalankan simulasi lengan robot kita. Kita gunakan perintah berikut:"

roslaunch seven_dof_arm_config demo.launch

- Penjelasan: "Perintah ini memulai simulasi dan memuat konfigurasi dasar dari robot. Pastikan semua paket ROS telah diinstal dengan benar, termasuk seven_dof_arm_config."

- Visual: Tampilan simulasi muncul di Rviz.

[Scene 2: Menambahkan Objek Kolisi]

- Visual: Terminal dan jendela Rviz.
- Narasi: "Langkah berikutnya adalah menambahkan objek kolisi ke lingkungan simulasi kita. Gunakan perintah ini:"

rosrun seven_dof_arm_test add_collision_object

- Penjelasan: "Script ini menambahkan objek seperti kotak atau silinder ke dalam simulasi. Objek-objek ini digunakan untuk menguji kemampuan robot dalam mendeteksi dan menghindari hambatan."
- Visual: Jendela Rviz memperlihatkan kotak muncul di area kerja robot.

[Scene 3: Melakukan Tugas Pick and Place]

- Visual: Terminal dan jendela Rviz.
- Narasi: "Sekarang kita siap untuk mengoperasikan robot dalam tugas pick and place. Jalankan perintah ini:"

rosrun seven_dof_arm_test pick_place

- Penjelasan: "Script ini menginstruksikan robot untuk mengambil objek dari satu lokasi dan memindahkannya ke lokasi lain. Pastikan bahwa konfigurasi jalur dan posisi target telah diatur dengan benar dalam kode."
- Visual: Animasi di Rviz yang menunjukkan robot mengambil dan memindahkan objek.

[Scene 5: Kesimpulan]

- Visual: Pembawa acara berbicara ke kamera.
- Narasi: "Itu tadi langkah-langkah untuk menjalankan simulasi robot lengan dengan ROS. Jangan lupa untuk memastikan bahwa semua dependensi telah diinstal dengan benar. Terima kasih telah menonton, dan sampai jumpa di video berikutnya!"