

Tutorial Menggunakan ROS MoveIt! dan Navigation Stack

[Intro]

(Visual: Logo ROS, MoveIt!, dan Navigation Stack dengan musik latar)

Narasi: "Halo semuanya! Selamat datang di tutorial kami tentang penggunaan ROS MoveIt! dan Navigation Stack. Dalam video ini, kita akan belajar bagaimana mengintegrasikan MoveIt! untuk perencanaan gerakan dan Navigation Stack untuk navigasi robot secara otonom. Yuk, kita mulai!"

[Segment 1: Persiapan Lingkungan]

(Visual: Komputer dengan terminal ROS terbuka)

Narasi: "Sebelum kita mulai, pastikan Anda telah menginstal ROS Noetic atau ROS 2 Foxy dan paket-paket MoveIt! serta Navigation Stack. Jika belum, ikuti panduan instalasi di situs resmi ROS."

(Visual: Terminal dengan perintah instalasi)

Narasi: "Gunakan perintah berikut untuk menginstal MoveIt! di ROS Noetic:

```
sudo apt-get install ros-noetic-moveit
```

Dan untuk Navigation Stack:

```
sudo apt-get install ros-noetic-navigation
```

"

[Segment 2: Konfigurasi MoveIt!]

(Visual: RViz dengan antarmuka MoveIt! terbuka)

Narasi: "Langkah pertama adalah membuat konfigurasi MoveIt! untuk robot Anda. Gunakan MoveIt! Setup Assistant untuk mempermudah proses ini."

(Visual: Menjalankan MoveIt! Setup Assistant di terminal)

Narasi: "Jalankan perintah berikut untuk membuka MoveIt! Setup Assistant:

```
roslaunch moveit_setup_assistant setup_assistant.launch
```

Di sini, Anda dapat memuat file URDF robot Anda, membuat grup gerakan, dan mengatur pengendalian gerakan."

[Segment 3: Perencanaan Gerakan dengan MoveIt!]

(Visual: RViz menunjukkan robot bergerak sesuai perencanaan)

Narasi: "Setelah konfigurasi selesai, jalankan MoveIt! untuk memulai perencanaan gerakan. Gunakan RViz untuk memvisualisasikan gerakan robot."

(Visual: Terminal menjalankan MoveIt!)

Narasi: "Gunakan perintah ini:

```
roslaunch my_robot_moveit_config demo.launch
```

Di RViz, Anda bisa menetapkan target gerakan dan melihat bagaimana robot merencanakan jalur terbaik."

[Segment 4: Mengintegrasikan Navigation Stack]

(Visual: Peta lingkungan dan robot bergerak secara otonom)

Narasi: "Selanjutnya, kita akan menggunakan Navigation Stack untuk navigasi otonom. Langkah pertama adalah membuat peta lingkungan menggunakan SLAM."

(Visual: Terminal menjalankan SLAM)

Narasi: "Jalankan perintah berikut untuk memulai SLAM:

```
roslaunch turtlebot3_slam turtlebot3_slam.launch
```

Setelah peta selesai dibuat, simpan peta tersebut dengan perintah:

```
roslaunch map_server map_saver -f my_map
```

"

(Visual: Robot bergerak menggunakan peta yang sudah disimpan)

Narasi: "Kemudian, gunakan peta ini untuk navigasi. Jalankan Navigation Stack dengan perintah berikut:

```
roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=my_map.yaml
```

"

[Segment 5: Pengujian dan Integrasi]

(Visual: Robot bergerak ke tujuan yang ditentukan)

Narasi: "Sekarang kita dapat mengintegrasikan MoveIt! dengan Navigation Stack. Gunakan MoveIt! untuk perencanaan gerakan robot lengan, dan Navigation Stack untuk mengarahkan robot mobile ke tujuan. Pastikan semua node ROS berjalan dengan lancar."

(Visual: Robot mobile dan robot lengan bekerja bersamaan)

Narasi: "Anda bisa menggunakan script Python untuk mengontrol gerakan secara bersamaan. Contoh sederhana untuk mengirim target navigasi adalah sebagai berikut:

```
from move_base_msgs.msg import MoveBaseAction, MoveBaseGoal
import rospy
import actionlib

def move_to_goal(x, y, z):
    client = actionlib.SimpleActionClient('move_base', MoveBaseAction)
    client.wait_for_server()
    goal = MoveBaseGoal()
    goal.target_pose.header.frame_id = 'map'
    goal.target_pose.pose.position.x = x
    goal.target_pose.pose.position.y = y
    goal.target_pose.pose.orientation.w = z
    client.send_goal(goal)
    client.wait_for_result()

move_to_goal(1.0, 2.0, 1.0)

"
```
