

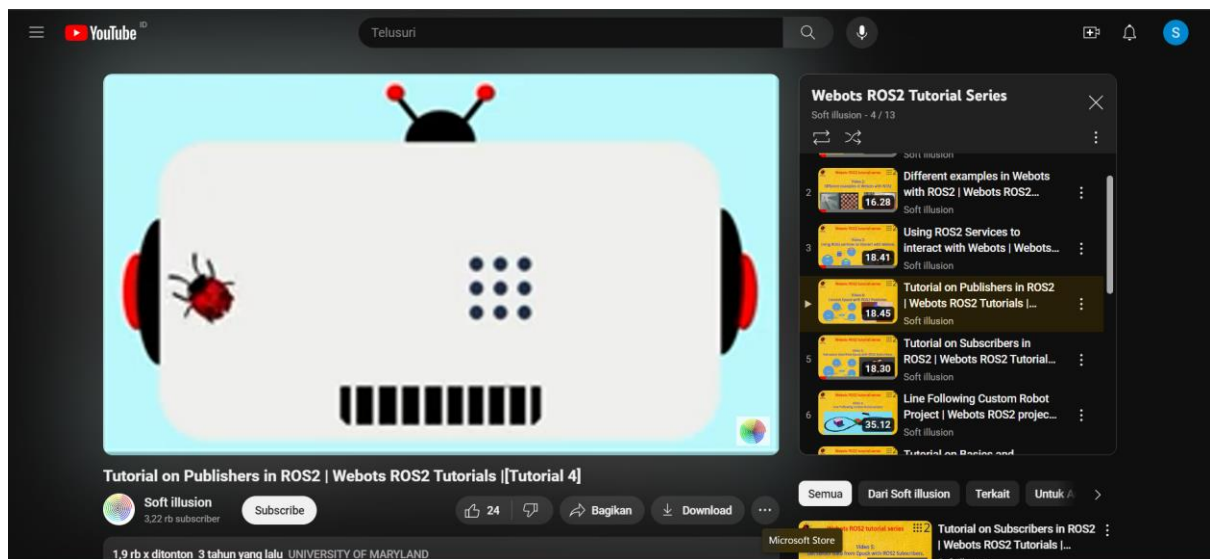
Syifa Wanda Isnaini

1103201248

Robotik

## Rangkuman Video 4 dan 5

### Video 4



Dalam video keempat dari seri tutorial Webster, teman-teman ilusi lembut membahas konsep penerbit dalam ROS 2. Mereka memulai dengan menjelaskan bahwa penerbit adalah node yang bertanggung jawab untuk mengirimkan pesan ke suatu topik dalam sistem ROS 2. Penerbit dapat ditulis dalam berbagai bahasa, seperti C++, Python, dan lainnya.

Video memberikan gambaran visual yang baik tentang konsep penerbit, menjelaskan peran utamanya dalam komunikasi antar node dalam ROS 2. Mereka kemudian melanjutkan dengan implementasi penerbit menggunakan terminal, mengajarkan cara mempublikasikan pesan ke suatu topik dengan perintah ROS 2.

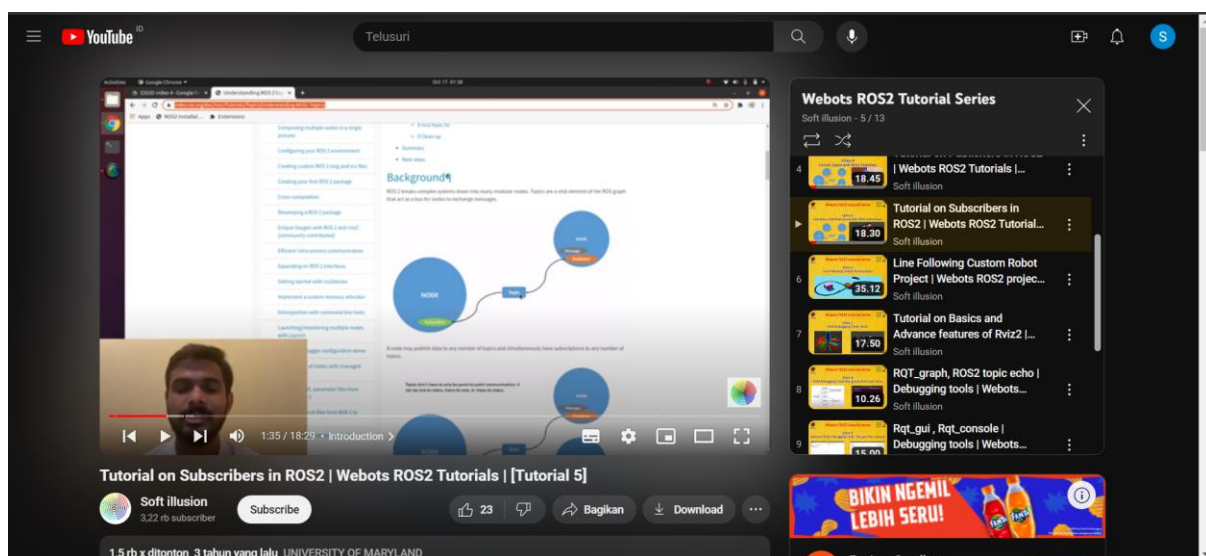
Selanjutnya, teman-teman ilusi lembut memberikan tutorial tentang cara membuat penerbit menggunakan skrip Python. Mereka menunjukkan langkah-langkah pembuatan paket, perubahan file `setup.py`, dan pembuatan kelas penerbit. Video juga mencakup cara menjalankan penerbit Python ini untuk mengirim pesan ke suatu topik.

Dalam bagian selanjutnya, video membahas perbedaan antara penerbit ROS 1 dan ROS 2. Beberapa perbedaan mencakup tipe pesan yang konsisten dalam semua bahasa di ROS 2, kemampuan beberapa node bertahan tanpa skor mentah, dan penggunaan DSS independen vendor.

Terakhir, teman-teman ilusi lembut membahas aplikasi penerbit dalam konteks robotika. Mereka menggambarkan pentingnya penerbit dalam mengendalikan sistem secara real-time, memperbarui informasi lingkungan, komunikasi internal sistem, dan mengatasi situasi darurat.

Video ini memberikan pemahaman yang baik tentang konsep penerbit dalam ROS 2, baik dari segi teori maupun implementasi praktisnya. Pemirsa diajak untuk memahami peran kunci penerbit dalam ekosistem ROS 2 dan bagaimana mengimplementasikannya dalam pengembangan robotika.

## Video 5



Dalam video kelima dari seri tutorial Webster, teman-teman ilusi lembut membahas konsep pelanggan dalam ROS 2. Mereka memulai dengan mengingatkan pemirsa untuk menonton video sebelumnya tentang cara membuat penerbit, karena topik dalam video ini akan terkait erat dengan konsep penerbit.

Video memberikan tautan ke halaman yang akan dibahas dan memberikan gambaran umum tentang apa yang akan dipelajari dalam video ini. Mereka menjelaskan bahwa pelanggan adalah node yang mengambil informasi dari suatu topik, dan bisa ada beberapa node yang berlangganan ke satu topik, atau satu node yang berlangganan banyak topik.

Implementasi pelanggan dari terminal digunakan untuk menggambarkan cara berlangganan topik dengan menggunakan robot epoch dan perintah `ros2 topic echo`. Mereka menunjukkan bagaimana mendapatkan nilai dari topik sensor TOF yang mempublikasikan jarak terhadap objek di depannya.

Selanjutnya, video membahas cara membuat node pelanggan menggunakan skrip Python. Mereka menunjukkan kode untuk membuat node pelanggan yang berlangganan topik sensor TOF, dan memberikan penjelasan langkah demi langkah.

Pembahasan selanjutnya melibatkan bagaimana menambahkan node pelanggan ke dalam file `setup.py` untuk menginisialisasi node saat membangun paket. Mereka menunjukkan langkah-langkah untuk memastikan node pelanggan dapat diakses dan dijalankan.

Dalam demonstrasi praktis, teman-teman ilusi lembut menggunakan robot epoch dan node pelanggan yang telah dibuat untuk mengamati perubahan nilai sensor TOF saat robot bergerak mendekati objek. Mereka juga memberikan instruksi langkah demi langkah tentang cara melakukan implementasi ini.

Terakhir, video menyajikan perbedaan antara pelanggan ROS 1 dan ROS 2. Beberapa perbedaan yang dibahas meliputi paradigma komunikasi, penggunaan DDS (layanan distribusi data), dan konsep kualitas layanan di ROS 2 yang memungkinkan pengaturan kebijakan seperti keandalan dan parameter riwayat.

Video ini memberikan pemahaman yang baik tentang konsep pelanggan dalam ROS 2, baik dari segi teori maupun implementasi praktisnya. Pemirsa diajak untuk memahami peran kunci pelanggan dalam ekosistem ROS 2 dan bagaimana mengimplementasikannya dalam pengembangan robotika.