

Naskah

## Chapter 1

Halo Guys saya Daanish, disini saya akan memperkenalkan apa itu ROS. **ROS** (Robot Operating System) adalah sebuah *framework* perangkat lunak sumber terbuka yang dirancang untuk membantu pengembangan aplikasi robotika. ROS adalah platform yang menyediakan alat, pustaka, dan konvensi untuk memudahkan pengembangan perangkat lunak robot.

Terdapat beberapa fitur utama ROS, yaitu

### Komunikasi Antar-Node

- ROS menggunakan arsitektur berbasis *publish-subscribe* untuk komunikasi antar komponen atau program (*nodes*). Komponen ini dapat berbagi data tanpa saling tergantung secara langsung.
- Contoh: Kamera robot mengirim data ke node pengolahan gambar.

### Modularitas

- ROS memungkinkan pengembang memecah perangkat lunak robot menjadi modul-modul kecil (*nodes*), sehingga lebih mudah untuk diuji, diperbaiki, dan dikembangkan.

### Pustaka dan Alat Bantu

- ROS menyediakan pustaka untuk berbagai tugas robotika, seperti kontrol motor, perencanaan jalur, dan pemrosesan data sensor.
- Contoh alat bantu: rviz (visualisasi) dan rqt (interface grafis).

### Kompatibilitas Multi-Bahasa

- ROS mendukung berbagai bahasa pemrograman seperti Python, C++, dan Lisp, sehingga memudahkan pengembang dengan latar belakang berbeda.

### Dukungan untuk Berbagai Platform

- ROS dapat digunakan pada robot nyata maupun dalam simulasi seperti Gazebo atau RViz.

### Sumber Terbuka

- ROS adalah proyek sumber terbuka yang dikembangkan oleh komunitas besar, sehingga tersedia dokumentasi, contoh kode, dan forum dukungan yang luas.

Lalu untuk komponen utamanya sendiri antara lain ada

### Master Node (roscore)

- Node pusat yang mengelola komunikasi antar node dalam jaringan ROS.

### Nodes

- Komponen atau program individu dalam aplikasi robot. Setiap node memiliki tugas spesifik, seperti membaca sensor atau mengendalikan aktuator.

## Topics

- Saluran komunikasi yang digunakan node untuk berbagi data.
- Contoh: Node kamera mempublikasikan gambar ke topik /camera.

## Services

- Mekanisme untuk interaksi dua arah antara node, seperti meminta data atau mengirim perintah.

## Action

- Seperti *services*, tetapi memungkinkan operasi yang berjalan dalam waktu lama dengan pembaruan status secara terus-menerus.

## Messages

- Format data yang digunakan untuk komunikasi antar node.

## Parameter Server

- Penyimpanan terpusat untuk konfigurasi parameter aplikasi.

Lalu terdapat beberapa versi ros yaitu **Noetic** (untuk Ubuntu 20.04). , **Melodic** (untuk Ubuntu 18.04). , **Foxy** dan **Humble** untuk ROS 2 (generasi berikutnya ROS).

Kapan kita menggunakan ROS?

ROS sangat cocok digunakan untuk: Pengembangan robot secara modular, Penelitian di bidang robotika, Prototipe robot cepat, Simulasi robot sebelum implementasi di perangkat keras nyata.

## Chapter 2

Halo saya Daanish, pada video ini saya akan menjelaskan bagaimana cara membuat ROS package

# 1. Pastikan Anda Sudah di Workspace ROS

# Ganti direktori ke workspace ROS Anda

```
cd ~/catkin_ws/src || {  
    echo "Workspace ROS tidak ditemukan. Pastikan Anda berada di direktori catkin_ws/src."  
    exit 1;  
}
```

# 2. Buat Paket Baru

```
# Ganti "my_package" dengan nama paket Anda
catkin_create_pkg my_package std_msgs rospy roscpp
```

### # 3. Verifikasi Struktur Paket

```
# Pastikan paket baru berhasil dibuat
echo "Paket ROS telah dibuat. Berikut adalah isi direktori paket:"
ls -l my_package
```

### # 4. Edit File CMakeLists.txt

```
# Instruksi ini memberikan informasi tentang menambahkan dependensi ke CMakeLists.txt
cd my_package || exit
```

### # 5. Tambahkan Dependensi Ekstra (Jika Diperlukan)

```
# Anda dapat menambahkan lebih banyak pustaka atau dependensi di CMakeLists.txt
nano CMakeLists.txt
```

### # 6. Edit File Package.xml

```
# Masukkan informasi tentang paket, seperti deskripsi dan dependensi tambahan
nano package.xml
```

### # 7. Kembali ke Workspace

```
cd ~/catkin_ws || exit
```

### # 8. Kompilasi Workspace ROS

```
catkin_make
```

### # 9. Sumberkan Setup File

```
# Ini memastikan ROS mengenali paket baru
source devel/setup.bash
```

### # 10. Verifikasi Paket

# Periksa apakah paket baru dikenali oleh ROS

```
rospack list | grep my_package
```

# 11. Jalankan Node atau Script dari Paket

# Tambahkan node Python atau C++ di dalam direktori src dari paket Anda

# Misalnya:

```
cd ~/catkin_ws/src/my_package/src || exit
```

```
nano my_script.py
```

# Tambahkan kode Python Anda, simpan file, lalu buat file dapat dieksekusi

```
chmod +x my_script.py
```

# Jalankan node Anda dengan:

```
roslaunch my_package my_script.py
```

# Catatan: Ubah nama "my\_package" dan "my\_script.py" sesuai kebutuhan Anda.

# Selesai!

## Chapter 3

```
#!/bin/bash
```

# Naskah untuk Mengatur dan Menjalankan ROS 1 dengan MoveIt dan RViz

# 1. Jalankan roscore untuk Memulai ROS Master

```
echo "Menjalankan roscore..."
```

```
roscore &
```

# 2. Buat Workspace ROS Baru

# Membuat direktori workspace

```
mkdir -p ~/catkin_ws/src
```

# 3. Arahkan Direktori ke Workspace

```
cd ~/catkin_ws/
```

# 4. Kompilasi Workspace

```
catkin_make
```

# 5. Atur Lingkungan Bash

# Tambahkan setup file ke .bashrc agar lingkungan ROS otomatis diatur

```
if ! grep -q "source ~/catkin_ws/devel/setup.bash" ~/.bashrc; then
```

```
    echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

```
    echo "Lingkungan ROS ditambahkan ke .bashrc"
```

```
else
```

```
    echo "Lingkungan ROS sudah ada di .bashrc"
```

```
fi
```

# Terapkan perubahan pada bashrc

```
source ~/.bashrc
```

# 6. Instal RViz dan MoveIt

# Pastikan dependensi visualisasi terinstal

```
echo "Menginstal RViz dan MoveIt..."
```

```
sudo apt update
```

```
sudo apt install -y ros-noetic-rviz ros-noetic-moveit
```

# 7. Verifikasi Instalasi

# Periksa apakah RViz dapat dijalankan

```
echo "Verifikasi instalasi dengan menjalankan RViz..."
```

```
rviz &
```

# Selesai

```
echo "Konfigurasi selesai. ROS 1, RViz, dan MoveIt siap digunakan."
```

## Chapter 4

```
#!/bin/bash
```

```
# Naskah untuk Menginstal dan Membuka Gazebo dengan ROS 1
```

```
# 1. Update Sistem
```

```
echo "Memperbarui daftar paket..."
```

```
sudo apt update
```

```
# 2. Instal Paket ROS Noetic dan Gazebo
```

```
echo "Menginstal ROS Noetic Desktop Full dan Gazebo..."
```

```
sudo apt install -y ros-noetic-desktop-full ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control
```

```
# 3. Verifikasi Instalasi dengan Membuka Gazebo
```

```
echo "Membuka Gazebo untuk verifikasi instalasi..."
```

```
gazebo &
```

```
# Selesai
```

```
echo "Instalasi selesai. Gazebo telah berhasil diinstal dan dijalankan."
```

Gazebo adalah alat yang sangat penting untuk pengembangan, pengujian, dan penelitian di bidang robotika. Dengan fitur simulasi dunia nyata, sensor, dan integrasi dengan ROS, Gazebo memungkinkan pengembang untuk membuat prototipe robot secara efisien sebelum diterapkan di dunia nyata.