

UAS Robotika

Mastering ROS for Robotics Programming: Bab I – Bab VI

BAB I - Pengenalan ROS

Dalam bab ini, kita akan mempelajari dasar-dasar Robot Operating System (ROS), termasuk alasan utama penggunaan ROS, struktur sistem file ROS, hierarki komputasi, dan komunitas ROS.

1.1 Alasan Menggunakan ROS

ROS menjadi pilihan utama dalam pengembangan sistem robotika modern karena beberapa alasan berikut:

- **Terbaru dan Potensial:** ROS adalah platform yang selalu diperbarui dan memiliki potensi besar dalam bidang robotika.
- **Alat dan Fungsi yang Lengkap:** ROS menyediakan berbagai alat dan fungsi yang memudahkan dalam visualisasi, pemrograman, dan simulasi robot.
- **Dukungan untuk Sensor Kelas Atas:** ROS sangat kompatibel dengan berbagai jenis sensor canggih yang digunakan dalam robotika.

1.2 Struktur Sistem File ROS

ROS memiliki struktur file sistem yang terorganisir dalam beberapa level, yaitu:

- **Metapackages:** Kumpulan paket virtual yang biasanya tidak berisi kode atau file, tetapi mengelompokkan paket-paket terkait.
- **Packages:** Elemen sentral dalam ROS yang berisi semua komponen yang diperlukan untuk suatu proyek, seperti kode, pesan, layanan, dan lainnya. Paket-paket ini diatur dalam urutan yang mencakup:
 - **Package Manifest:** Berisi informasi tentang paket, seperti penulis, lisensi, dependensi, dan flag kompilasi.
 - **Messages:** Struktur data yang digunakan untuk komunikasi antar-node.
 - **Services:** Mekanisme untuk permintaan dan respons data antar-node.
 - **Code:** Kode sumber yang mengimplementasikan fungsionalitas paket.
 - **Miscellaneous:** Berkas-berkas lainnya yang mendukung paket.

1.3 Hierarki Komputasi di ROS

Komputasi di ROS diatur dalam beberapa komponen utama, yaitu:

- **Nodes:** Proses individu yang melakukan tugas komputasi spesifik.
- **Master:** Entitas yang menangani registrasi nama dan pencarian proses (nodes) lain.
- **Parameter Server:** Penyimpanan terpusat untuk parameter yang dapat diakses oleh nodes.
- **Messages:** Data yang dikirimkan antar-nodes melalui topik.
- **Topics:** Saluran komunikasi di mana messages dipublikasikan dan disubscrib.
- **Services:** Layanan yang memungkinkan komunikasi sinkron antar-nodes melalui permintaan dan respons.
- **Bags:** Format file untuk menyimpan dan memutar ulang messages.

BAB II – *Getting Started with ROS Programming*

Fase Pertama dalam Pembuatan Robot: Desain dan Pemodelan

Fase awal dalam pembuatan robot mencakup proses desain dan pemodelan. Dalam tahap ini, kita menggunakan alat CAD (*Computer-Aided Design*) seperti Autodesk Fusion 360, SolidWorks, Blender, dan lainnya untuk merancang dan memodelkan robot. Tujuan utama dari pemodelan robot adalah untuk melakukan simulasi, yang memungkinkan kita untuk mengidentifikasi dan memperbaiki cacat kritis dalam desain sebelum melanjutkan ke fase produksi.

1.1 Tujuan Pemodelan dan Simulasi

Simulasi robotik adalah langkah penting karena:

- **Memastikan Fungsi Desain:** Simulasi membantu memastikan bahwa desain robot berfungsi dengan baik sesuai dengan spesifikasi yang diinginkan.
- **Identifikasi Cacat:** Membantu mengidentifikasi dan memperbaiki cacat desain sebelum masuk ke fase produksi yang lebih mahal dan memakan waktu.

1.2 Proyek Desain Robot

Dalam bab ini, kita akan membahas proses desain dua jenis robot:

- **Manipulator dengan Tujuh Derajat Kebebasan (DOF):** Robot yang dirancang untuk memiliki tujuh titik artikulasi atau sendi, memungkinkan fleksibilitas gerak yang tinggi.
- **Robot dengan Penggerak Diferensial:** Robot yang menggunakan dua roda yang digerakkan secara independen untuk navigasi dan manuver.

1.3 Tahapan Lanjutan

Pada bab-bab selanjutnya, kita akan melanjutkan dengan:

- **Simulasi:** Menggunakan perangkat lunak simulasi untuk menguji model robot.
- **Pembangunan Perangkat Keras:** Membangun robot fisik berdasarkan desain yang telah disimulasikan.
- **Interfacing dengan ROS:** Mengintegrasikan robot dengan ROS untuk pengendalian dan pemantauan.

1.4 Menggunakan ROS untuk Desain dan Pemodelan

Jika kita berencana membuat model 3D robot dan mensimulasikannya menggunakan ROS, kita perlu mempelajari beberapa paket ROS yang dapat membantu dalam proses ini.

Membuat model robot di ROS penting untuk berbagai alasan:

- **Simulasi dan Kontrol:** Untuk mensimulasikan dan mengontrol robot secara virtual.
- **Visualisasi:** Untuk memvisualisasikan robot dalam lingkungan simulasi.
- **Informasi Struktur dan Kinematika:** Menggunakan alat ROS untuk mendapatkan informasi mengenai struktur dan kinematika robot.

ROS menyediakan beberapa paket penting untuk mendesain dan membuat model robot, di antaranya:

- **urdf:** Digunakan untuk membuat deskripsi model 3D robot dalam format XML.
- **kdl_parser:** Untuk memarsing deskripsi kinematika robot.
- **robot_state_publisher:** Untuk mempublikasikan transformasi keadaan robot berdasarkan model URDF.
- **collada_urdf:** Untuk mengonversi model URDF ke format COLLADA, yang dapat digunakan dalam berbagai aplikasi simulasi.

Paket-paket ini membantu kita dalam membuat deskripsi model 3D robot yang sesuai dengan karakteristik perangkat keras sebenarnya, memungkinkan simulasi yang lebih akurat dan pengendalian yang efektif.

*note: terdapat masalah (untuk seluruh bab ini) yang dialami yaitu ada error dalam “mastering_ros_demo_pkg” karena paket tersebut “*is not found*”.

The screenshot shows a Linux desktop environment with a sidebar of application icons on the left. The primary monitor displays a web browser window showing the '1.5 Environment setup' page from the 'docker-compose-v3' repository on GitHub. The browser's address bar shows 'https://www.github.com/docker/compose-v3'. The page content includes instructions for environment setup, such as installing Docker and setting up environment variables. The secondary monitor displays the '1.6 Dependencies for building packages' page from the same repository, which lists dependencies for building packages and provides instructions for installing them. The browser's address bar also shows 'https://www.github.com/docker/compose-v3'.

The screenshot displays three terminal windows in a Kali Linux environment. The leftmost window shows the user installing the 'j0k3r' script from a GitHub repository and making it executable. The middle window shows the user running the script, which initiates a listener on port 1337. The rightmost window shows the script being executed on a remote machine (192.168.1.101), which then connects back to the listener on port 1337. A red box highlights the 'Start recording', 'Pause recording', and 'Finish recording' buttons in the top right corner of the rightmost terminal window.

[illegible]

Ini merupakan contoh setelah mengikuti bagian action server dan client*:

```

$ rosrun demo_server demo_server
INFO [162608097.000000]: Starting demo action server
INFO [162608097.000000]: demo_action is processing the goal 10
INFO [162608097.000000]: Setting to goal 0 / 10
INFO [162608097.000000]: Setting to goal 1 / 10
INFO [162608097.000000]: Setting to goal 2 / 10
INFO [162608097.000000]: Setting to goal 3 / 10
INFO [162608097.000000]: Setting to goal 4 / 10
INFO [162608097.000000]: demo_action got prompted!

$ rosrun demo_client demo_client
INFO [162608097.000000]: Waiting for action server to start.
INFO [162608097.000000]: Action server started. Sending goal.
INFO [162608097.000000]: Sending goal [10] and timeout time of [1]
INFO [162608097.000000]: Action file not finish before the time out.
demo_client[162608097.000000]:

```

BAB III - Working with ROS for 3D Modelling

Dalam bab ini, kita akan mempelajari cara menggunakan ROS (*Robot Operating System*) untuk pemodelan 3D robot. Fokus utama adalah memahami dan menggunakan *Unified Robot Description Format* (URDF) serta alternatifnya, yaitu Xacro.

1.1 Pengenalan URDF

URDF (*Unified Robot Description Format*) adalah format file yang digunakan untuk mendeskripsikan model 3D robot dalam ROS. File URDF berisi informasi tentang geometri, kinematika, dan properti fisik robot. Berikut adalah langkah-langkah umum dalam menggunakan URDF:

- **Membuat Model Robot:** Menulis file URDF yang mendeskripsikan robot.
- **Visualisasi di RVIZ:** Menggunakan RVIZ, aplikasi visualisasi di ROS, untuk melihat dan memeriksa model robot yang dibuat.

1.2 Tantangan dan Kendala

Sayangnya, akses ke GitHub untuk mendapatkan contoh dan kode URDF mungkin terbatas atau tidak tersedia. Namun, kita masih bisa merujuk pada buku dan sumber daya lain yang menyediakan kode dan panduan untuk membuat model robot.

1.3 Kekurangan URDF

Meskipun URDF adalah alat yang kuat, ada beberapa keterbatasan, antara lain:

- **Kemudahan Penggunaan:** URDF bisa jadi kompleks dan sulit digunakan untuk pemula.
- **Kemampuan untuk Digunakan Kembali:** URDF kurang fleksibel dalam hal penggunaan ulang kode.
- **Modularitas:** Kurang mendukung pembuatan model yang modular.
- **Kemudahan Pemrograman:** Penulisan dan pemeliharaan file URDF bisa menjadi rumit.

1.4 Pengenalan Xacro

Untuk mengatasi beberapa keterbatasan URDF, ROS menyediakan format alternatif yaitu Xacro (XML Macros). Xacro membantu menyederhanakan penulisan file URDF dengan menggunakan makro, sehingga lebih mudah untuk digunakan dan diprogram. Keunggulan Xacro meliputi:

- **Sederhana dan Mudah Digunakan:** Mempermudah penulisan dan pemeliharaan model robot.
- **Modularitas:** Memungkinkan pembuatan model yang lebih modular dan terorganisir.
- **Kemampuan untuk Digunakan Kembali:** Xacro memungkinkan penggunaan ulang kode yang lebih baik.

Setelah menulis file Xacro, kita dapat mengonversinya menjadi URDF jika diperlukan, sehingga tetap kompatibel dengan alat-alat lain di ROS.

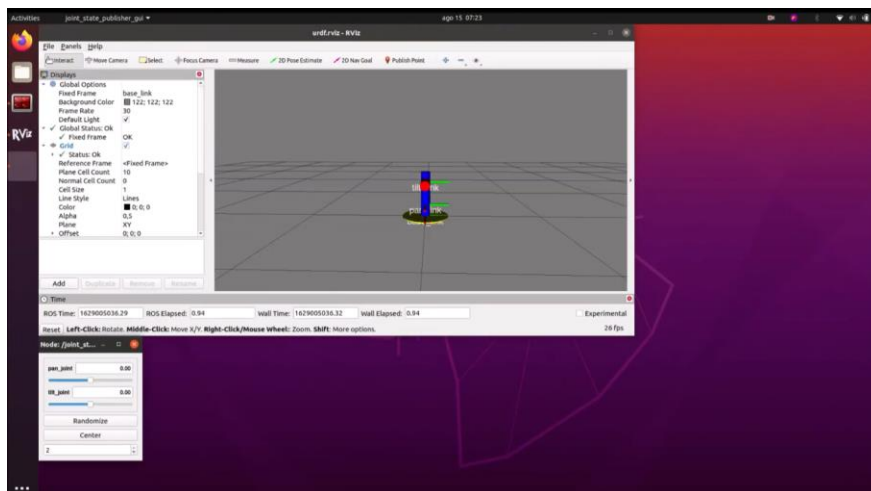
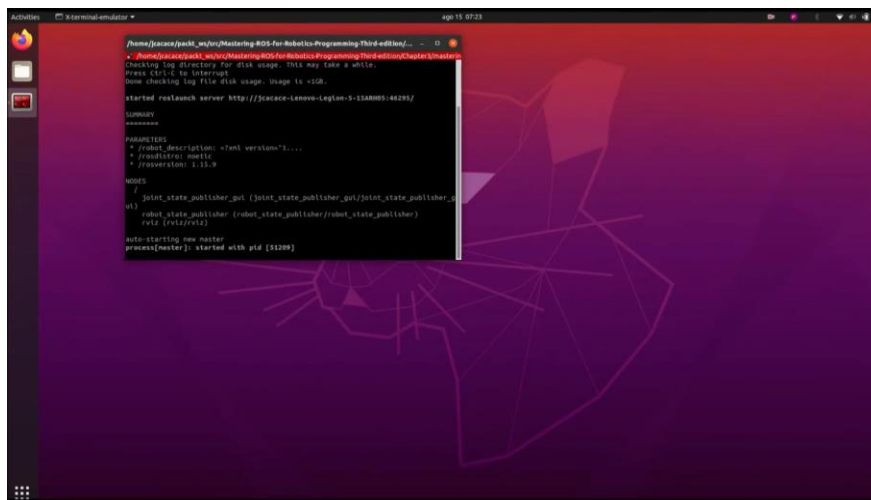
1.5 Implementasi dan Visualisasi

Langkah-langkah untuk menggunakan Xacro dalam pemodelan robot meliputi:

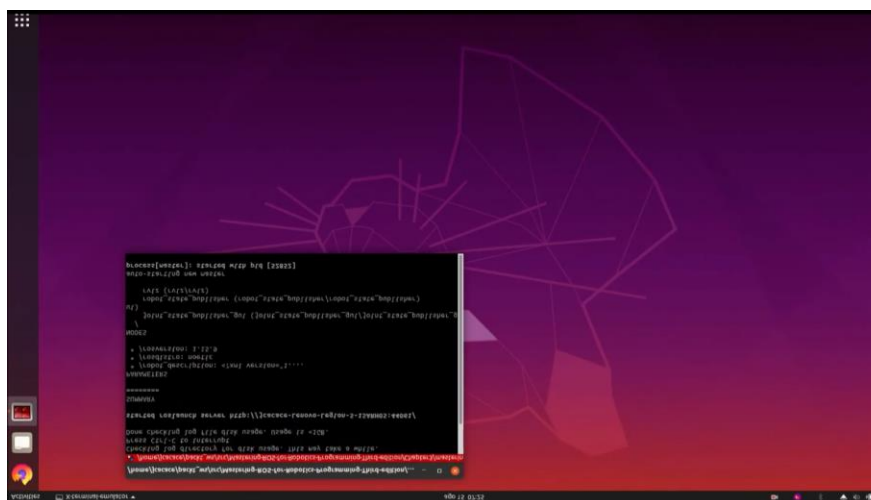
- **Membuat File Xacro:** Menulis deskripsi model robot dalam format Xacro.
- **Mengonversi ke URDF:** Menggunakan perintah ROS untuk mengonversi file Xacro menjadi URDF.
- **Visualisasi di RVIZ:** Memvisualisasikan model robot di RVIZ untuk memastikan desain dan fungsi yang diinginkan.

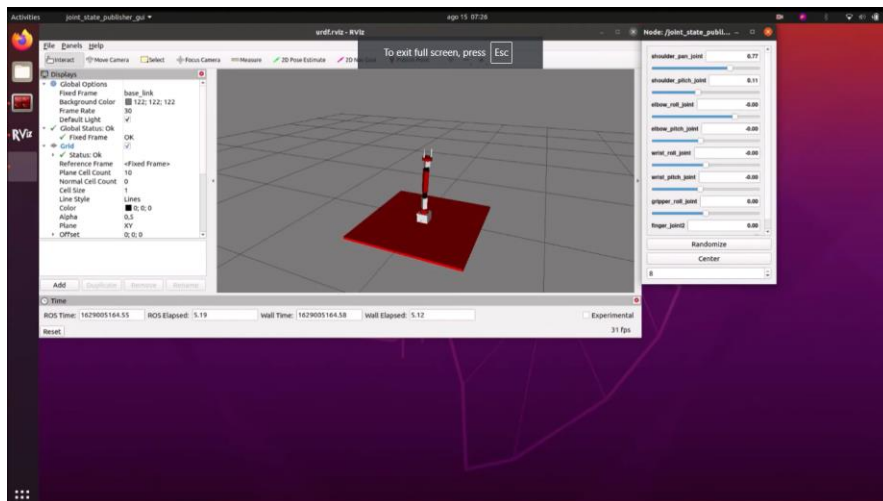
Untuk hasil contoh dari bab ini bisa dilihat berikut ini:

(1) Visualizing the 3D Robot model in Rviz

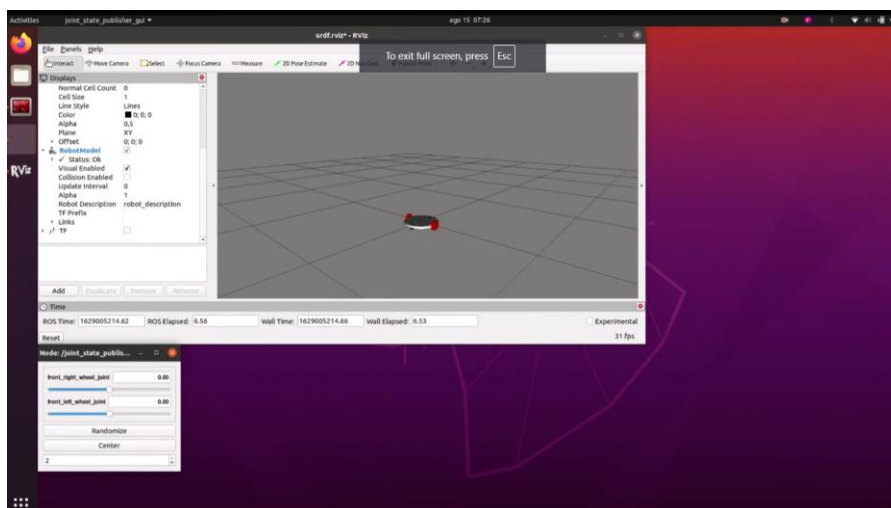
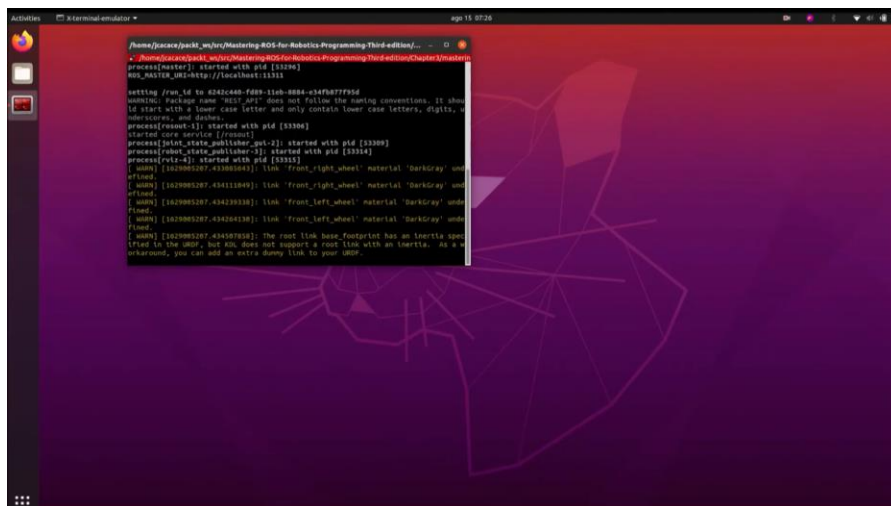


(2) Explaining the XARCO model of the seven-DOF arm





(3)Creating a Robot model for the differential drive mobile



BAB IV - *Simulating Robots Using ROS & Gazebo*

Dalam bab ini, kita akan mempelajari beberapa konsep lanjutan dalam Robot Operating System (ROS), yaitu ROS pluginlib, nodelets, dan plugin Gazebo. Kita akan mengeksplorasi fungsi dan aplikasi masing-masing konsep, serta melihat contoh untuk mendemonstrasikan cara kerjanya.

1.1 ROS Pluginlib

Pluginlib adalah kerangka kerja di ROS yang memungkinkan pembuatan dan penggunaan plugin. Plugin adalah modul yang dapat dimuat secara dinamis, memungkinkan fleksibilitas dan modularitas dalam pengembangan perangkat lunak ROS. Dengan pluginlib, kita dapat membuat komponen perangkat lunak yang bisa diperluas tanpa perlu mengubah kode dasar aplikasi.

1.2 ROS Nodelets

Nodelets adalah bentuk khusus dari ROS node yang memungkinkan beberapa node berjalan dalam satu proses. Keuntungan utama dari nodelets adalah mengurangi overhead komunikasi antarproses, yang bisa meningkatkan performa aplikasi ROS, terutama dalam kasus yang memerlukan komunikasi data yang cepat dan sering.

- **Keuntungan Nodelets:**
 - **Efisiensi Komunikasi:** Menghindari overhead komunikasi antarproses dengan berbagi memori antar-node.
 - **Penggunaan Sumber Daya yang Lebih Baik:** Mengurangi penggunaan CPU dan memori dibandingkan dengan menjalankan node terpisah.

1.3 Plugin Gazebo

Plugin Gazebo digunakan untuk memperluas fungsionalitas simulator Gazebo, yang sering digunakan bersama ROS untuk mensimulasikan robot dan lingkungan mereka. Plugin ini memungkinkan kita untuk menambahkan perilaku khusus, sensor, atau aktuator ke dalam simulasi Gazebo.

- **Penggunaan Plugin Gazebo:**
 - **Simulasi Sensor dan Aktuator:** Menambahkan dan mengkonfigurasi berbagai sensor dan aktuator pada model robot di dalam Gazebo.
 - **Perilaku Kustom:** Menerapkan logika kustom dan perilaku dinamis pada elemen simulasi.

1.4 Implementasi dan Contoh

Dalam bab ini, kita juga akan melihat cara membuat dan menggunakan plugin Gazebo serta nodelets dalam ROS:

- **Membuat Plugin Gazebo:** Langkah-langkah untuk membuat plugin yang dapat memperluas kemampuan simulasi di Gazebo.
- **Menggunakan Nodelets:** Contoh implementasi nodelets dan cara mengkonfigurasinya untuk meningkatkan performa aplikasi ROS.
- **Contoh Pluginlib:** Studi kasus penggunaan pluginlib untuk membuat sistem yang modular dan fleksibel.

Untuk hasil contoh dari bab ini bisa dilihat berikut ini:

(1) Moving the robot joints using ROS controllers in Gazebo

