*Technical Report from Mastering ROS*

***Final Examination Robot Autonomy***

A logo of a university

Description automatically generated

Oleh:

Nama : Rizky Ramadhani Syam

NIM : 1103204086

**PROGRAM STUDI TEKNIK KOMPUTER**

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

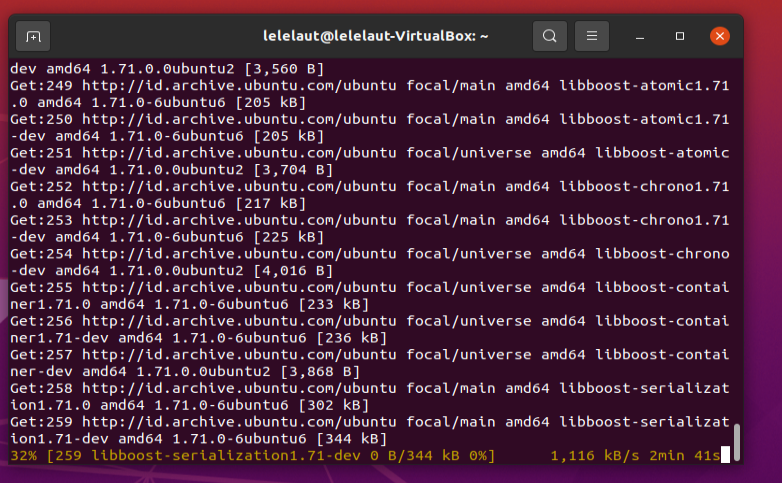
**2023**

***Chapter 1 – Introduction to ROS***

Instalisasi Awal

* Ubuntu 20.04 LTSD/Debian 10
* ROS Noetic

Panduan instalasi dapat ditemukan di tautan berikut : [*http://wiki.ros.org/noetic/Installation/Ubuntu*](http://wiki.ros.org/noetic/Installation/Ubuntu)*.*



Instalasi ROS

ROS

ROS, singkatan dari Robot Operating System, adalah sebuah framework perangkat lunak yang dirancang untuk mempermudah pengembangan aplikasi robotika. Meskipun disebut sebagai "sistem operasi", ROS sebenarnya lebih sebagai platform perangkat lunak yang berjalan di atas sistem operasi Linux. Ini memberikan beragam alat, perpustakaan, dan konvensi yang digunakan untuk mengelola komunikasi antar-komponen dalam sistem robotika.

Berikut adalah beberapa poin kunci tentang ROS:

1. **Komunikasi Antar-Komponen**: ROS menyediakan infrastruktur yang memungkinkan komponen atau node dalam sistem robotika berkomunikasi satu sama lain melalui topik, layanan, dan panggilan prosedur jarak jauh (RPC).
2. **Manajemen Paket**: ROS menggunakan sistem manajemen paket yang memungkinkan pengguna untuk dengan mudah menginstal, memperbarui, dan mengelola kode dan dependensinya.
3. **Library dan Tools**: ROS dilengkapi dengan berbagai perpustakaan dan alat yang memudahkan pengembangan aplikasi robotika, termasuk navigasi, penginderaan, penglihatan komputer, manipulasi, dan lainnya.
4. **Multi-Platform**: Meskipun awalnya dikembangkan untuk Linux, ROS telah dikembangkan untuk mendukung beberapa platform seperti macOS, Windows, dan bahkan sistem operasi berbasis real-time.
5. **Komunitas yang Kuat**: ROS memiliki komunitas yang besar dan aktif yang menyediakan banyak dokumentasi, tutorial, dan paket yang siap digunakan.

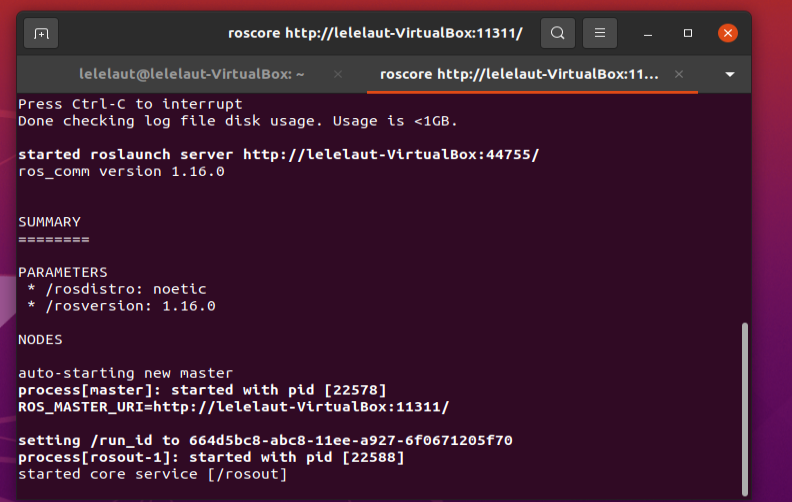
ROS digunakan secara luas dalam pengembangan robotika, dari penelitian akademis hingga aplikasi industri. Karena fleksibilitasnya, ROS memungkinkan para pengembang untuk fokus pada pengembangan aplikasi robotika tanpa harus memulai dari awal dalam membangun infrastruktur komunikasi yang kompleks.

Sebelum mulai menggunakan node-node ROS, perlu untuk mengaktifkan ROS Master dan ROS Parameter Server. Ini bisa dilakukan dengan perintah tunggal yang disebut **roscore**. Perintah ini akan memulai beberapa program kunci:

* **ROS Master**: Bertanggung jawab atas pengaturan komunikasi antar-node di lingkungan ROS.
* **ROS Parameter Server**: Tempat penyimpanan parameter yang bisa diakses oleh node-node dalam sistem ROS.
* **Node logging rosout**: Sebuah node yang mengumpulkan dan menyimpan pesan log dari node-node ROS lainnya. Ini meneruskan pesan-pesan log ini ke topik yang lain untuk penggunaan lebih lanjut.

Node **rosout** ini mengambil pesan log dari node-node ROS menggunakan perpustakaan klien ROS seperti roscpp dan rospy. Kemudian, pesan-pesan ini diteruskan ke topik yang disebut **/rosout\_agg**, yang merupakan tempat terkumpulnya pesan log. Penting untuk diingat bahwa **roscore** perlu dijalankan sebelum memulai node-node ROS apapun.

Jadi, **roscore** adalah perintah yang krusial sebelum memulai penggunaan sistem ROS. Itu memastikan ROS Master, Parameter Server, dan mekanisme logging siap digunakan.



Roscore

Kesimpulan

Framework perangkat lunak ROS saat ini semakin populer di kalangan ahli robotika. Pengetahuan tentang ROS menjadi semakin penting bagi mereka yang berencana membangun karir sebagai insinyur robotika dalam beberapa tahun mendatang. Dalam bab ini, dasar-dasar ROS telah dibahas, terutama dengan tujuan menyegarkan pemahaman konsep jika sebelumnya Anda sudah familiar dengan ROS. Pentingnya mempelajari ROS dan keunggulannya di antara platform perangkat lunak robotika saat ini telah dibahas. Konsep dasar seperti ROS master dan parameter server telah dijelajahi, serta penjelasan tentang cara kerja roscore telah diberikan. Pada bab berikutnya, manajemen paket ROS akan diperkenalkan dan beberapa contoh praktis dari sistem komunikasi ROS akan dibahas.

***Chapter 2 – Getting Started with ROS Programming***

Membuat Paket ROS

Paket ROS merupakan unit dasar dari program-program ROS. Kita bisa membuat paket ROS, membangunnya, dan merilisnya ke publik. Distribusi ROS yang kita gunakan saat ini adalah Noetic Ninjemys. Kita menggunakan sistem build catkin untuk membangun paket-paket ROS. Sistem build bertanggung jawab dalam menghasilkan target (eksekutabel/pustaka) dari kode sumber teks yang dapat digunakan oleh pengguna akhir. Pada distribusi ROS yang lebih lama seperti Electric dan Fuerte, rosbuild adalah sistem build yang digunakan. Namun, karena berbagai kelemahan dari rosbuild, catkin kemudian diciptakan. Hal ini juga memungkinkan kita untuk mendekatkan sistem kompilasi ROS kepada Cross Platform Make (CMake). Ini memiliki banyak keunggulan, seperti kemampuan untuk memindahkan paket ke sistem operasi lain, misalnya Windows. Jika sebuah sistem operasi mendukung CMake dan Python, paket-paket berbasis catkin dapat dipindahkan ke sistem tersebut.

A screenshot of a computer

Description automatically generated

Catkin make

A screenshot of a computer

Description automatically generated

Catkin create pkg

Membuat ROS Node

***Chapter* 3 – *Working with ROS for 3D Modeling***

Dalam bab ini, kita akan membahas proses desain dari dua robot. Salah satunya adalah manipulator dengan tujuh Derajat-of-Freedom (DOF), dan yang lainnya adalah robot penggerak differential drive. Di bab-bab berikutnya, kita akan melihat simulasi, mempelajari cara membangun perangkat keras nyata, dan membahas antarmuka dengan ROS.

ROS menyediakan beberapa paket untuk merancang dan membuat model robot, seperti urdf, kdl\_parser, robot\_state\_publisher, dan collada\_urdf. Paket-paket ini akan membantu kita dalam membuat deskripsi model robot 3D dengan karakteristik yang sama persis dengan perangkat keras nyata.

Kita dapat mendefinisikan model robot, sensor, dan lingkungan kerja menggunakan URDF. Kita juga dapat mengurai (parse) mereka menggunakan parser URDF. Kita hanya dapat mendeskripsikan robot dalam URDF yang memiliki struktur seperti pohon pada link-linknya, yang berarti robot akan memiliki link yang kaku dan terhubung menggunakan joint (sambungan). Link yang fleksibel tidak dapat direpresentasikan menggunakan URDF. URDF terdiri dari tag-tag XML khusus, dan kita dapat mengurai tag-tag XML ini menggunakan program parser untuk pengolahan lebih lanjut.

A screenshot of a computer

Description automatically generated

Contoh URDF menggunakan Rviz

Sekarang, kita dapat membuat beberapa robot kompleks menggunakan URDF dan xacro. Robot pertama yang akan kita buat adalah sebuah lengan robotik tujuh-DOF, yang merupakan manipulator rangkaian serial dengan beberapa link serial. Lengan tujuh-DOF ini memiliki kelebihan kinematika, yang berarti memiliki lebih banyak sendi dan DOF daripada yang diperlukan untuk mencapai posisi dan orientasi tujuannya. Keuntungan dari manipulator yang redundan adalah bahwa kita dapat memiliki lebih banyak konfigurasi sendi untuk posisi dan orientasi tujuan yang diinginkan. Ini akan meningkatkan fleksibilitas dan kecanggihan pergerakan robot serta dapat menerapkan gerakan bebas-tabrakan yang efektif di ruang kerja robotik.

A computer screen shot of a computer program

Description automatically generated

***Kesimpulan***

Dalam bab ini, terutama dibahas pentingnya pemodelan robot dan bagaimana robot dapat dimodelkan di ROS. Pembahasan dilakukan mengenai paket-paket yang digunakan di ROS untuk memodelkan struktur robot, seperti urdf, xacro, dan joint\_state\_publisher beserta antarmukanya. URDF, xacro, dan tag-tag UTDF utama yang dapat digunakan juga dibahas. Model contoh dalam URDF dan xacro dibuat, dan perbedaan di antara keduanya dibahas. Selanjutnya, manipulator robotik kompleks dengan tujuh Derajat Kebebasan (DOF) dibuat, dan penggunaan paket joint\_state\_publisher dan robot\_state\_publisher dipelajari. Pada akhir bab, prosedur desain robot mobile penggerak differensial menggunakan xacro ditinjau. Pada bab berikutnya, simulasi robot ini akan dipelajari menggunakan Gazebo.

***Chapter* 4 – *Simulating Robots Using ROS and Gazebo***

Setelah mendesain model 3D dari sebuah robot, tahap berikutnya adalah mensimulasikannya. Simulasi robot akan memberikan gambaran tentang bagaimana robot beroperasi dalam lingkungan virtual.

Kita akan menggunakan simulator Gazebo (<http://www.gazebosim.org/>) untuk mensimulasikan lengan robotik dengan tujuh Derajat Kebebasan (DOF) dan juga robot mobile. Gazebo merupakan simulator multi-robot untuk simulasi robotik kompleks di dalam maupun di luar ruangan. Kita dapat mensimulasikan robot kompleks, sensor-sensor robot, dan berbagai objek 3D. Gazebo sudah memiliki model-model simulasi dari robot-robot populer, sensor-sensor, dan beragam objek 3D di repositorinya (<https://bitbucket.org/osrf/gazebo_models/>). Kita bisa langsung menggunakan model-model ini tanpa harus membuat yang baru.

Gazebo terintegrasi dengan sempurna dalam ROS berkat antarmuka ROS yang memungkinkan kontrol lengkap terhadap Gazebo di dalam ROS. Kita bisa menginstal Gazebo tanpa ROS, tetapi sebaiknya menginstal antarmuka ROS-Gazebo untuk berkomunikasi dari ROS ke Gazebo.

Di bab ini, kita akan membahas simulasi lengan dengan tujuh DOF dan robot beroda differential. Kita juga akan membahas pengontrol ROS yang membantu mengendalikan sendi-sendi robot di Gazebo.

*A screenshot of a computer

Description automatically generated*

DOF Robot in Gazebo

Selanjutnya kita bisa mengendalikan sendi sendi dari robot ini menggunakan controller ROS. Simulasi lengan di Gazebo diinisiasi oleh file-file peluncuran, dan konfigurasi pengontrol, pengontrol status sendi, dan pengontrol posisi sendi dimuat. Selanjutnya, dijalankan penerbit status robot, yang bertanggung jawab untuk memublikasikan status sendi dan transformasi (TF)

kita dapat mulai memberikan perintah kepada setiap sendi untuk mencapai posisi yang diinginkan. Untuk menggerakkan sebuah sendi robot di Gazebo, kita perlu memublikasikan nilai sendi yang diinginkan dengan tipe pesan std\_msgs/Float64 ke topik perintah pengontrol posisi sendi.

Mari kita lihat bagaimana kontroler ROS berinteraksi dengan Gazebo. Gambar berikut menunjukkan hubungan antara kontroler ROS, antarmuka perangkat keras robot, dan simulator/perangkat keras nyata:

A diagram of a system

Description automatically generated

Kita dapat melihat alat-alat pihak ketiga, paket navigasi, dan MoveIt!. Paket-paket ini dapat memberikan tujuan (set point) kepada kontroler robot mobile dan kontroler lengan robotik. Kontroler-kontroler ini dapat mengirimkan posisi, kecepatan, atau usaha ke antarmuka perangkat keras robot.

***Summary***

Dalam bab ini, upaya dilakukan untuk mensimulasikan dua robot: yang pertama adalah sebuah lengan robot dengan tujuh derajat kebebasan (DOF), dan yang lainnya adalah sebuah robot beroda differential. Dimulai dengan lengan robot, diskusi dilakukan mengenai tag Gazebo tambahan yang diperlukan untuk meluncurkan robot di Gazebo. Pembahasan juga mencakup cara menambahkan sensor visi 3D ke dalam simulasi. Kemudian, sebuah file peluncuran dibuat untuk memulai Gazebo dengan lengan robot, dan dibahas bagaimana menambahkan pengontrol ke setiap sendi. Pengontrol ditambahkan dan bekerja dengan setiap sendi. Seperti halnya dengan lengan robot, URDF untuk simulasi Gazebo dibuat dan plugin Gazebo-ROS yang diperlukan untuk pemindai laser dan mekanisme penggerak roda differential ditambahkan. Setelah menyelesaikan model simulasi, simulasi diluncurkan menggunakan file

***Chapter* 5 – *Simulating Robots Using ROS, CoppeliaSim, and Webots***

Setelah mempelajari cara mensimulasikan robot dengan Gazebo, dalam bab ini kita akan membahas dua perangkat lunak simulasi robot lainnya yang sangat kuat: CoppeliaSim (http://www.coppeliarobotics.com) dan Webots (https://cyberbotics.com/).

Kedua simulasi ini adalah simulator robot multiplatform. CoppeliaSim dikembangkan oleh Coppelia Robotics. Ia menawarkan banyak model simulasi robot industri dan mobile yang populer yang siap digunakan, serta berbagai fungsionalitas yang dapat dengan mudah diintegrasikan dan dikombinasikan melalui antarmuka pemrograman aplikasi (API) yang khusus. Selain itu, CoppeliaSim dapat beroperasi dengan Robot Operating System (ROS) menggunakan antarmuka komunikasi yang tepat, yang memungkinkan kita mengontrol adegan simulasi dan robot melalui topik dan layanan. Seperti halnya dengan Gazebo, CoppeliaSim bisa digunakan sebagai perangkat lunak mandiri, namun sebuah plugin eksternal harus diinstal untuk bekerja dengan ROS.

Sedangkan Webots adalah perangkat lunak sumber terbuka yang digunakan untuk mensimulasikan robot 3D. Perangkat lunak ini dikembangkan oleh Cyberbotics Ltd. dan sejak Desember 2018 telah dirilis di bawah lisensi sumber terbuka dan gratis.

Sebelum mulai bekerja dengan CoppeliaSim, kita perlu menginstalnya di sistem kita dan mengonfigurasi lingkungan kita untuk memulai jembatan komunikasi antara ROS dan adegan simulasi. CoppeliaSim adalah perangkat lunak lintas-platform, tersedia untuk berbagai sistem operasi seperti Windows, macOS, dan Linux. Ini dikembangkan oleh Coppelia Robotics GmbH dan didistribusikan dengan lisensi edukasi gratis maupun lisensi komersial. Unduh versi terbaru dari simulator CoppeliaSim dari halaman unduhan Coppelia Robotics di http://www.coppeliarobotics.com/downloads.html, pilih versi edu untuk Linux. Di bab ini, kami akan merujuk pada versi CoppeliaSim 4.2.0.

A screenshot of a computer

Description automatically generated

Setelah mengunduh, instal CoppeliaSim sesuai instruksi yang disediakan oleh pengembang. Selanjutnya, kita akan perlu mengatur jembatan komunikasi antara CoppeliaSim dan ROS agar keduanya dapat berinteraksi. Langkah-langkah ini umumnya akan melibatkan penggunaan plugin eksternal yang disediakan oleh Coppelia Robotics atau plugin ROS yang khusus untuk integrasi dengan CoppeliaSim.

A computer screen shot of a computer program

Description automatically generated

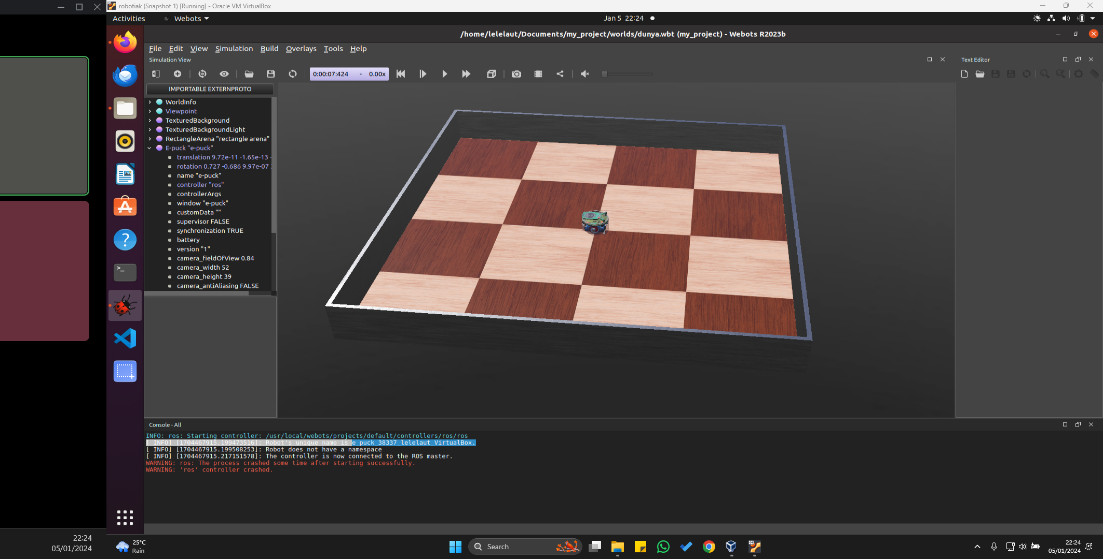
Tampilan Aplikasi CoppeliaSim

Sebagaimana yang telah dilakukan dengan CoppeliaSim, kita perlu menginstal Webots di sistem kita sebelum menyiapkannya dengan ROS. Webots adalah perangkat lunak simulasi multiplatform yang didukung oleh Windows, Linux, dan macOS. Perangkat lunak ini awalnya dikembangkan oleh Swiss Federal Institute of Technology, Lausanne (EPFL). Saat ini, Webots dikembangkan oleh Cyberbotics dan dirilis di bawah lisensi sumber terbuka Apache 2.

Webots menyediakan lingkungan pengembangan lengkap untuk memodelkan, memrogram, dan mensimulasikan robot. Ini dirancang untuk penggunaan profesional dan banyak digunakan dalam industri, pendidikan, dan riset.

Anda dapat memilih berbagai cara untuk menginstal simulator ini. Anda bisa mengunduh paket .deb dari halaman web Webots (http://www.cyberbotics.com/#download) atau menggunakan manajer paket Debian/Ubuntu Advanced Packaging Tool (APT).

Jika memilih untuk menggunakan paket .deb, Anda bisa mengunduhnya dari situs web Webots dan mengikuti instruksi penginstalannya. Alternatifnya, jika Anda menggunakan Debian atau Ubuntu, Anda bisa memanfaatkan APT untuk menginstal Webots secara lebih mudah dan terintegrasi dengan sistem Anda.gunakan APT untuk menginstal Webots dengan lebih mudah dan terintegrasi dengan sistem Anda.



Tampilan Aplikai webots

Kesimpulan

Dalam bab ini, kita utamanya mengulangi apa yang telah kita lakukan sebelumnya dalam bab sebelumnya dengan Gazebo, namun menggunakan simulator robot lainnya: CoppeliaSim dan Webots. Keduanya adalah program perangkat lunak simulasi multiplatform yang mengintegrasikan berbagai teknologi dan sangat fleksibel. Berkat antarmuka pengguna yang intuitif, mereka mungkin lebih mudah digunakan bagi pengguna baru.

Kita utamanya mensimulasikan dua robot, salah satunya diimpor menggunakan file URDF dari lengan tujuh-DOF yang telah dirancang dalam bab-bab sebelumnya, sedangkan yang lainnya adalah robot beroda differential populer yang disediakan oleh model simulasi Webots. Kita mempelajari cara menghubungkan dan mengendalikan sendi-sendi robot pada model kita dengan ROS, serta bagaimana menggerakkan robot mobile penggerak differential menggunakan topik-topik.

Di bab selanjutnya, kita akan melihat bagaimana menghubungkan lengan robotik dengan paket MoveIt ROS dan robot mobile dengan tumpukan navigasi.