Laporan Hasil Analisis Simulasi

Oleh Rafli Limandijaya/1103210243

Kelas TK45G09 Robotika

*Note : Pengambilan gambar yang saya lakukan diambil dari video penjelasan yang saya buat sendiri

1. Chapter 1 Introduction to ROS

   ROS adalah framework fleksibel yang menyediakan berbagai tools dan

   libraries untuk menulis software Robot robot. Beberapa kelebihan di ROS adalah :

   a. Kemampuan Tingkat Tinggi: ROS menyediakan fungsi siap pakai seperti SLAM, AMCL, dan MoveIt untuk navigasi otonom dan perencanaan gerak robot.
   b. Alat Pendukung: ROS dilengkapi alat kuat seperti rqt_gui, RViz, dan Gazebo untuk debugging, visualisasi, dan simulasi.
   c. Dukungan Sensor Canggih: ROS mendukung integrasi sensor seperti 3D LIDAR, pemindai laser, dan sensor kedalaman.
   d. Interoperabilitas: Node ROS dapat dibuat dalam berbagai bahasa yang mendukung pustaka klien ROS.
   e. Modularitas: Node yang terpisah memungkinkan sistem tetap berfungsi meski salah satu node crash.
   f. Penanganan Konkuren: ROS memungkinkan banyak node mengakses data yang sama, mengurangi kompleksitas dan meningkatkan debugging.

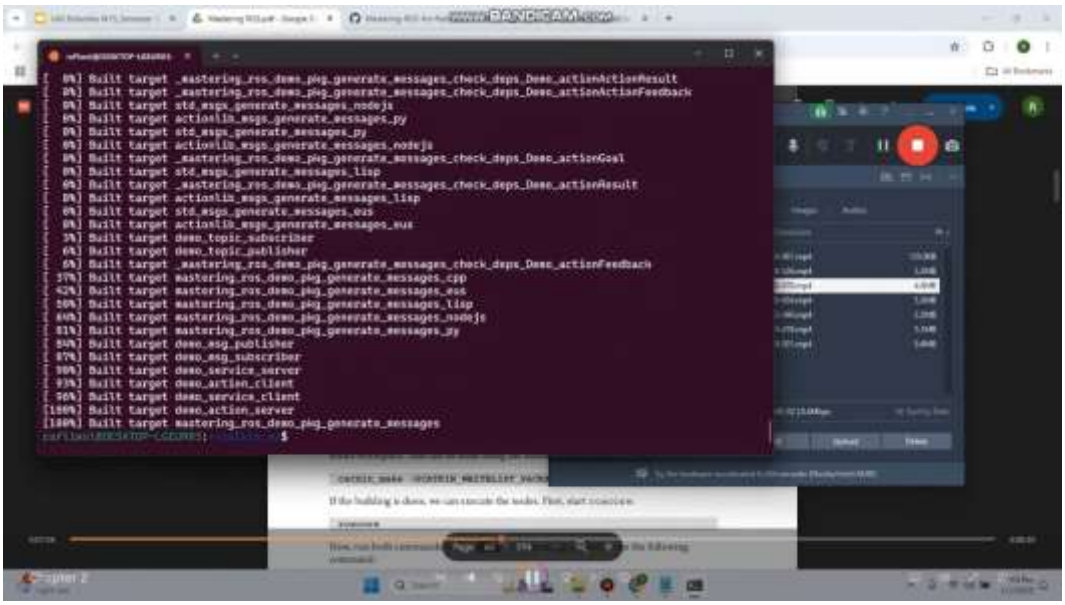
   Komputasi dalam ROS dilakukan menggunakan jaringan node ROS. Jaringan komputasi ini disebut computation graph. Konsep utama dalam computation graph meliputi node ROS, master, parameter server, messages, topics, services, dan bags. Setiap konsep dalam graph ini memiliki kontribusi yang

berbeda untuk membangun jaringan tersebut. Paket-paket yang terkait dengan komunikasi dalam ROS, termasuk pustaka klien inti seperti roscpp dan rospython, serta implementasi konsep seperti topics, nodes, parameters, dan services, disatukan dalam sebuah stack yang disebut ros_comm.

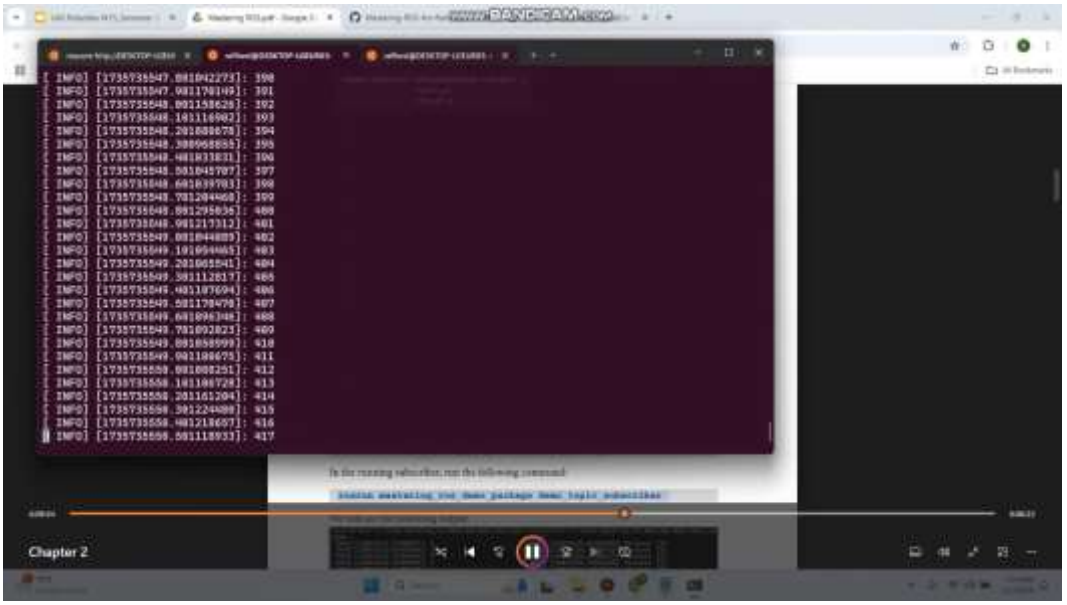
2. Getting Started with ROS Programming

Pembuatan workspace di ROS dilakukan melalui catkin workspace. Catkin Workspace adalah folder tempat pengguna untuk memodifikasi, membangun, dan menginstal paket catkin. Distribusi ROS yang saat ini kita gunakan adalah Noetic Ninjemys. Kita menggunakan sistem build catkin untuk membangun paket-paket ROS. Sistem build bertanggung jawab untuk menghasilkan target (berupa eksekusi/libraries) dari kode sumber dalam bentuk teks yang dapat digunakan oleh pengguna akhir.

Langkah pertama yang harus dilakukan adalah membuat sebuah catkin workspace dan juga memasukkan package ke folder src di dalam workspace. Command yang saya gunakan untuk membuat Catkin Workspace terdapat pada dokumen command saya.




Jika sudah, saatnya menjalankan command launch untuk subscriber dan publisher. Pada terminal pertama, jadikan sebagai core dari operasi, terminal kedua dan ketiga sebagai publisher dan subscriber.




Yang terjadi disini adalah publisher mengirimkan data integer ke core dan subscriber mengatakan kembali apa yang diberikan oleh publisher. Ada juga contoh kedua yang mensimulasikan jika message yang digunakan berupa custom message. Untuk penjelasan lebih lanjut, silakan tonton video yang telah saya buat.

3. Working with ROS for 3D Modeling

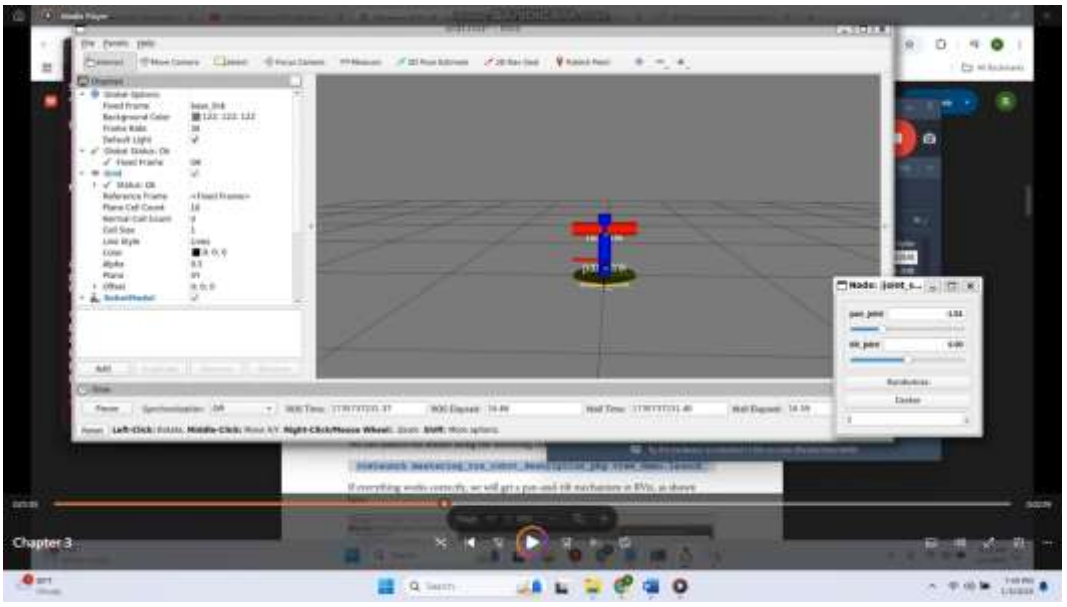
Simulasi di Chapter 3 dicontohkan menggunakan Rviz di ROS. Robot yang didesain bernama Seven DOF Robot dan Differential Drive Mobile Robot. Ada beberapa hal yang perlu kita pahami sebelum memulai modelling yaitu:

urdf: Paket ROS paling penting untuk memodelkan robot adalah urdf. Paket ini berisi parser C++ untuk URDF, yaitu file XML yang merepresentasikan model robot. Paket urdf terdiri dari beberapa komponen, antara lain:

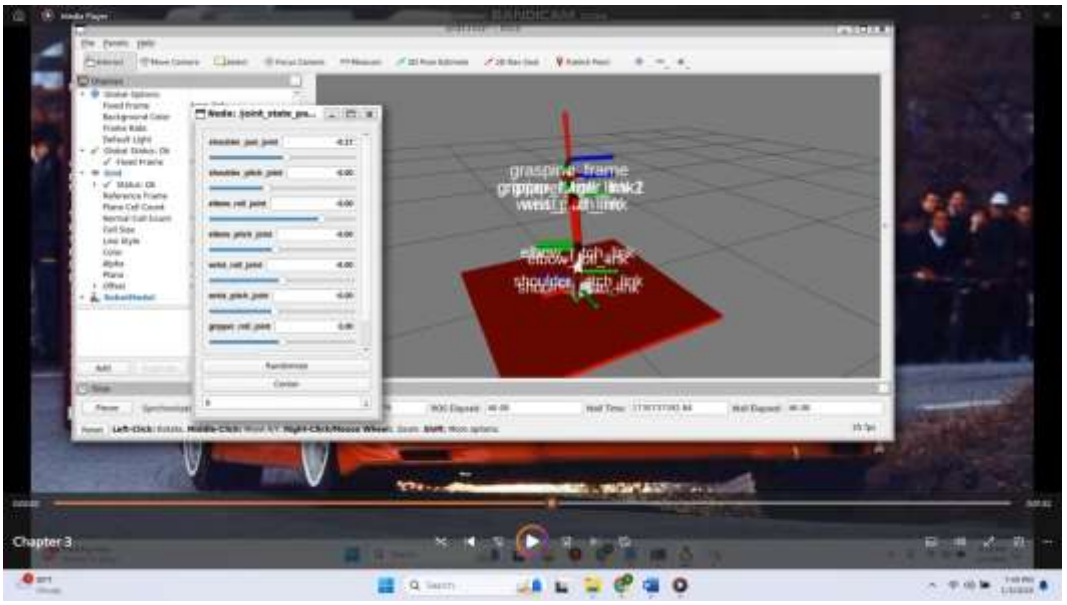
a. urdf_parser_plugin: Paket ini mengimplementasikan metode untuk mengisi struktur data URDF.
b. urdfdom_headers: Komponen ini menyediakan header struktur data inti untuk digunakan oleh parser URDF.
c. collada_parser: Paket ini mengisi struktur data dengan cara mem-parsing file Collada.
d. urdfdom: Komponen ini mengisi struktur data dengan mem-parsing file URDF.

Langkah yang harus dilakukan untuk mendesain robotnya adalah untuk membuat kode xml. Dalam simulasi yang saya lakukan, digunakan file desain dari repository github yang sudah ada. Untuk melihat hasil dari desain, bisa lakukan dengan command yang saya tulis pada file command.

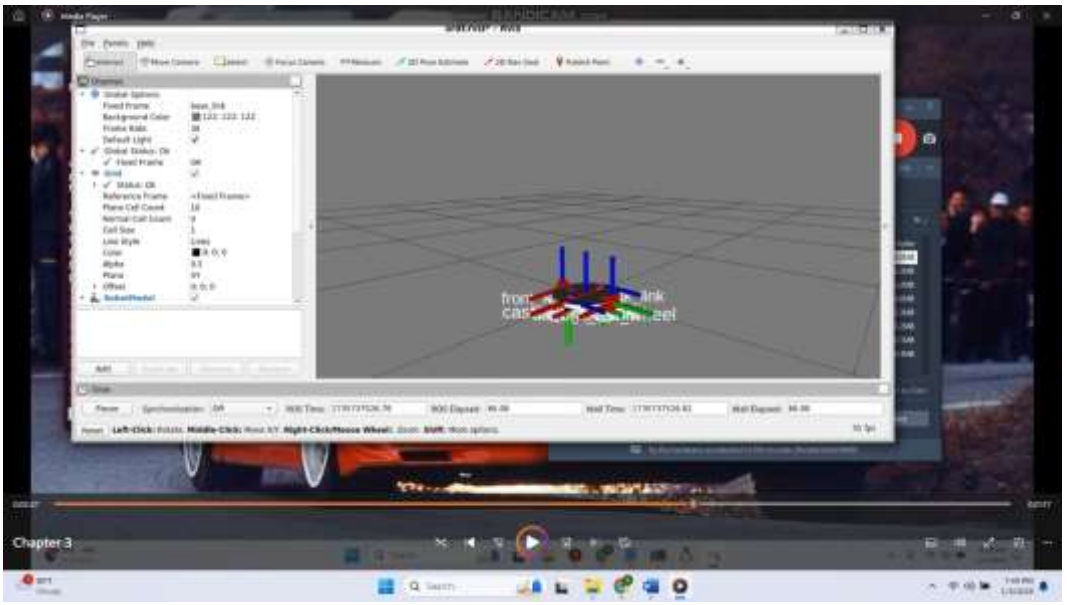
Robot pertama memiliki 2 joint yang dapat digerakkan, yaitu pan joint dan tilt joint.




Robot kedua, yaitu 7DOF Robot memiliki 7 joint yang bisa kita gerakkan.



Robot ketiga, yaitu Differential Drive Mobile Robot, memiliki 2 joint yang bisa digerakkan yaitu roda kanan dan kiri. Hal ini berguna untuk membuat masing-masing robot dapat bergerak ke arah yang diperlukan.

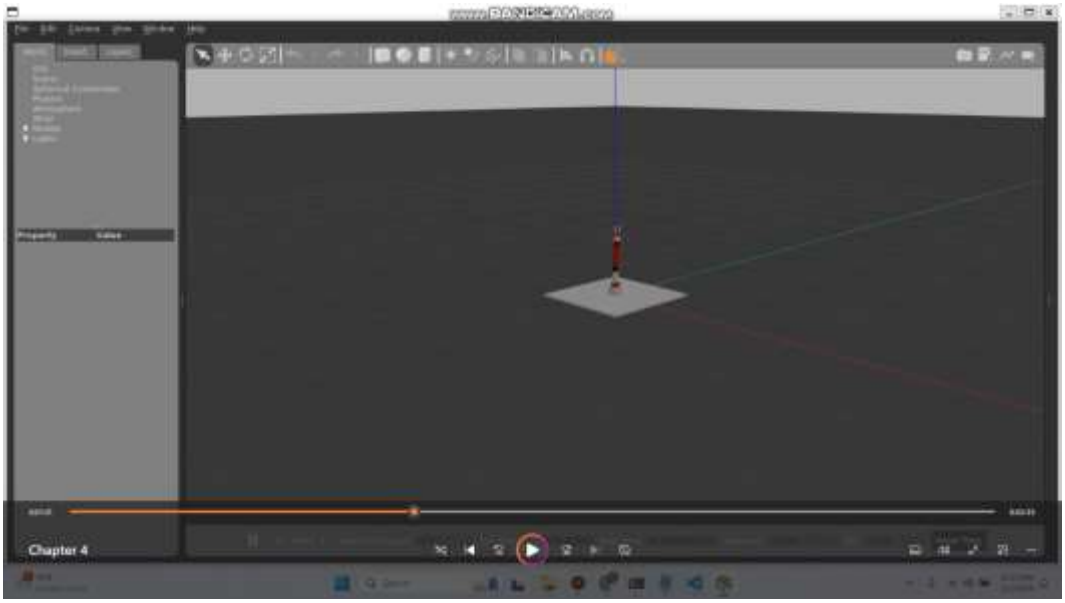


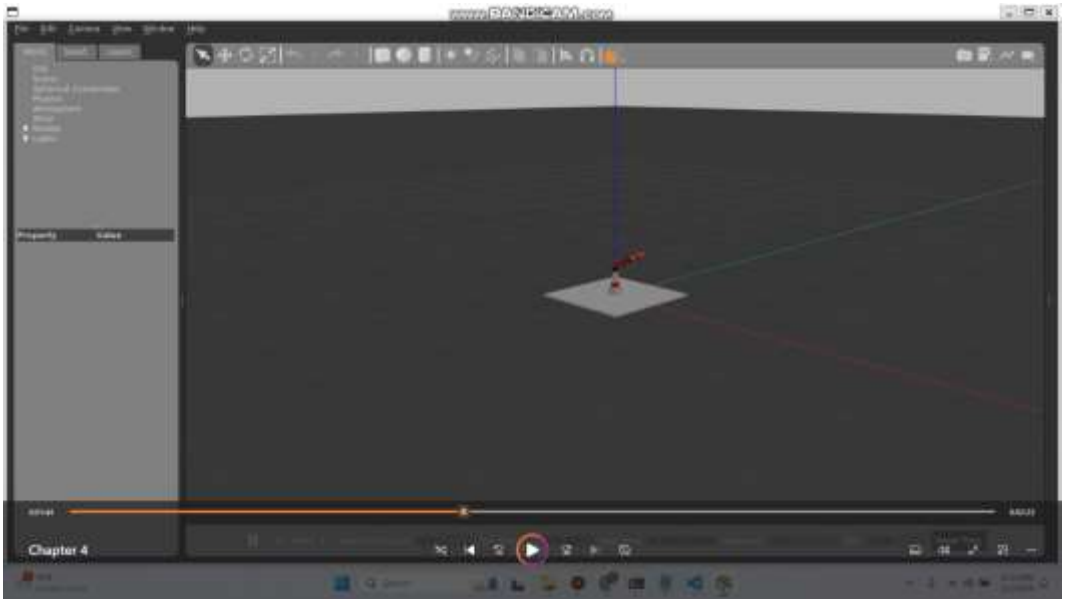

Secara umum Rviz digunakan untuk mengetes desain yang telah kita buat. Kita bisa menggerakan joint-joint yang telah kita buat dan melihat apakah hasilnya sesuai. Dalam robotika, hal tersebut disebut motion planning. Jika motion planning sudah sesuai, maka bisa disimulasikan di Gazebo.

4. Simulating Robots Using ROS and Gazebo

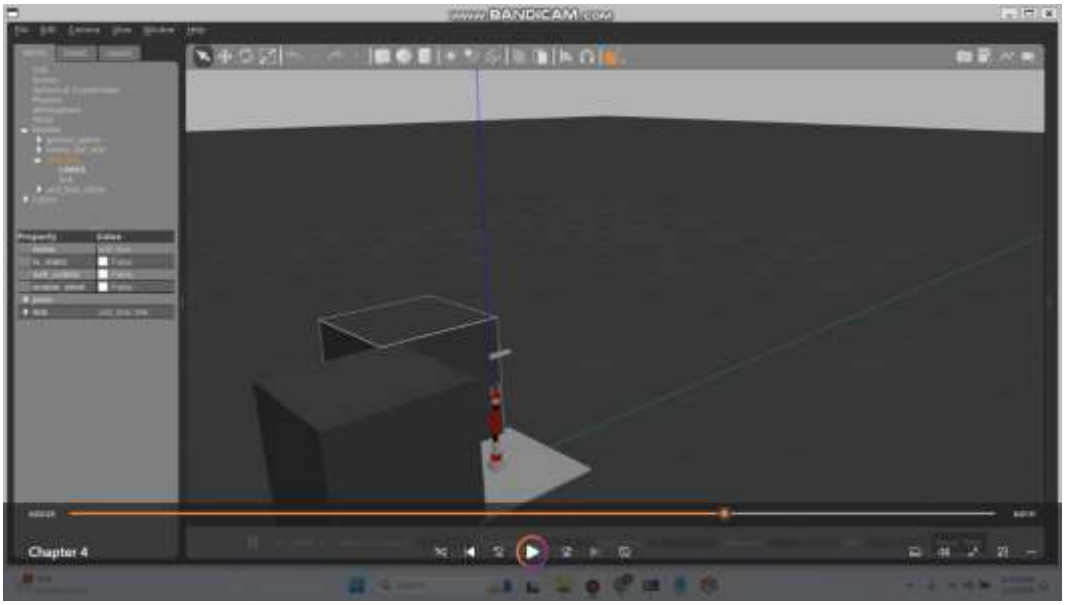
Untuk mensimulasikan desain dari robot yang sudah dibuat dan commandnya, dapat dilakukan di Gazebo. Gazebo adalah simulator multi-robot untuk simulasi robotik kompleks di lingkungan dalam dan luar ruangan. Dengan Gazebo, kita dapat mensimulasikan robot kompleks, sensor robot, dan berbagai objek 3D. Gazebo sudah memiliki model simulasi robot populer, sensor, dan berbagai objek 3D di repositorinya, sehingga kita bisa langsung menggunakannya tanpa perlu membuat model baru. Gazebo terintegrasi sempurna dengan ROS melalui antarmuka ROS yang memberikan kontrol penuh Gazebo di dalam ROS. Meskipun Gazebo dapat diinstal tanpa ROS, kita perlu menginstal antarmuka ROS-Gazebo untuk memungkinkan komunikasi antara ROS dan Gazebo.

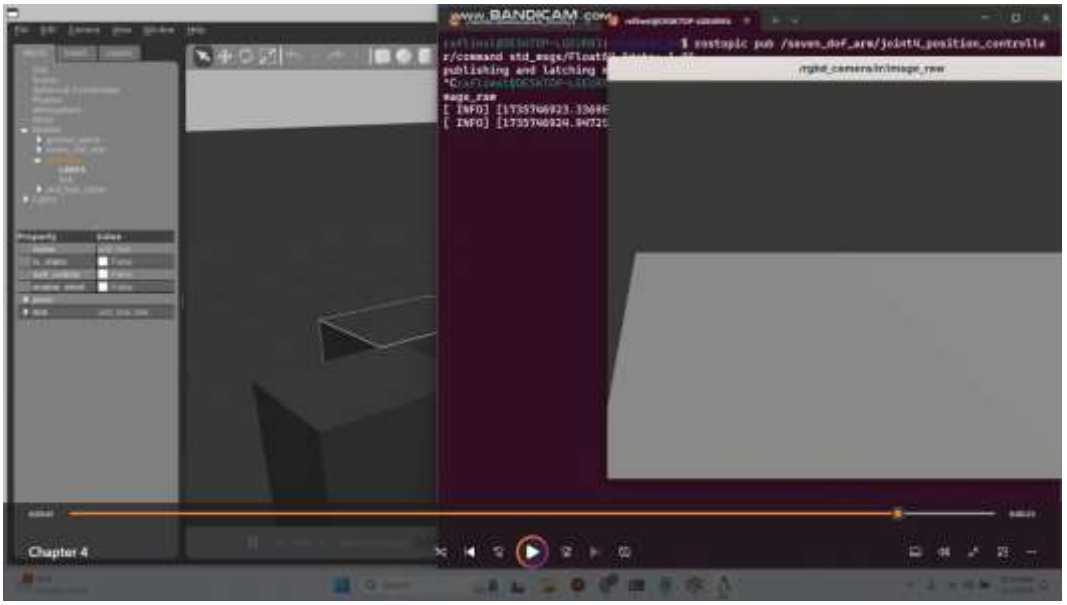
Untuk launching, dilakukan penulisan command seperti yang saya tulis di file command saya. Pada terminal pertama tuliskan command untuk membuka model dan pada terminal kedua tuliskan command untuk menggerakan joint robot. Saya menggerakan joint ke 4 ke arah 1.0, dan dapat dilihat robotnya berubah posisi dari tegak menjadi sedikit ke bawah pada joint ke 4.




Saya juga melakukan percobaan menggunakan robot yang sama, namun dengan sensor RGBD. Pada percobaan ini, saya mengambil gambar yang didapatkan oleh sensor tersebut.




Sekian hasil analisis yang saya lakukan untuk chapter 1-4 dari buku ROS for Robotics Programming Third Edition.