

Laporan UAS Robotika



ANALISIS E-BOOK “MASTERING ROS”

Chapter 1-4

Oleh:

Arya Putra Siahaan (1103213099)

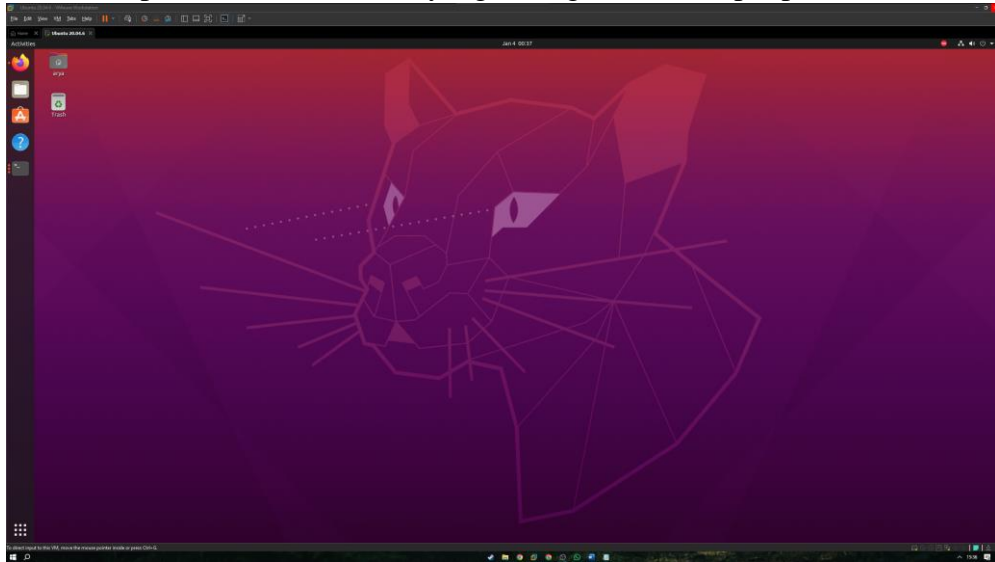
ROBOTIKA TK-45-G09

**PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK KOMPUTER
UNIVERSITAS TELKOM
BANDUNG**

2025

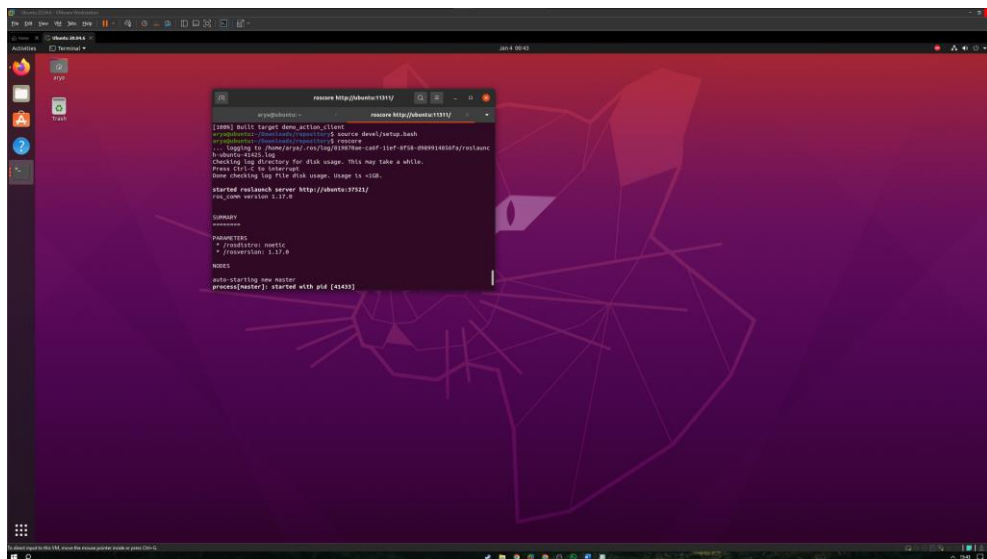
Dokumentasi Chapter 1

1. Mengunduh Virtual Machine pada perangkat yang akan anda gunakan seperti : VirtualBox, VMWare, Oracle, Blackbox
2. Mengunduh Ubuntu sesuai Versi yang akan digunakan pada Virtual machine : Saya menggunakan versi 20.04.6 pada virtual machine saya
3. Install Ubuntu pada Virtual Machine yang anda gunakan sampai proses akhir



Gambar 1.1 Ubuntu dan VMWare Berhasil Terinstall

4. Install ROS, Gunakan ROS Noetic dengan link berikut :
<https://wiki.ros.org/noetic/Installation/Ubuntu>
5. Ikuti Tutorial Instalasi diatas dan Pastikan berhasil menjalankan “roscore” di terminal Virtual Machine anda



Gambar 1.2 Roscore berhasil berjalan

Roscore berhasil berjalan berarti komponen inti ROS telah terinstall dan berfungsi dengan baik.

Analisis Chapter 1

Dokumentasi Chapter 1 menjelaskan langkah-langkah mempersiapkan lingkungan pengembangan ROS menggunakan mesin virtual dan Ubuntu. Dimulai dengan mengunduh perangkat lunak Virtual Machine seperti VirtualBox, VMware, atau lainnya untuk menyediakan lingkungan terisolasi yang ideal bagi eksperimen dan pengembangan tanpa mengubah sistem operasi asli. Setelah itu, diunduh ISO Ubuntu versi 20.04.6, yang merupakan distribusi Linux LTS yang didukung secara resmi oleh ROS Noetic.

Proses instalasi Ubuntu dilakukan di mesin virtual hingga sistem siap digunakan, dengan alokasi sumber daya yang memadai seperti minimal 4 GB RAM dan 25 GB penyimpanan disk. Selanjutnya, dilakukan instalasi ROS Noetic melalui panduan resmi di wiki.ros.org, mencakup pemasangan paket-paket inti seperti ros-noetic-desktop-full, konfigurasi rosdep, dan pengaturan variabel lingkungan dengan source.

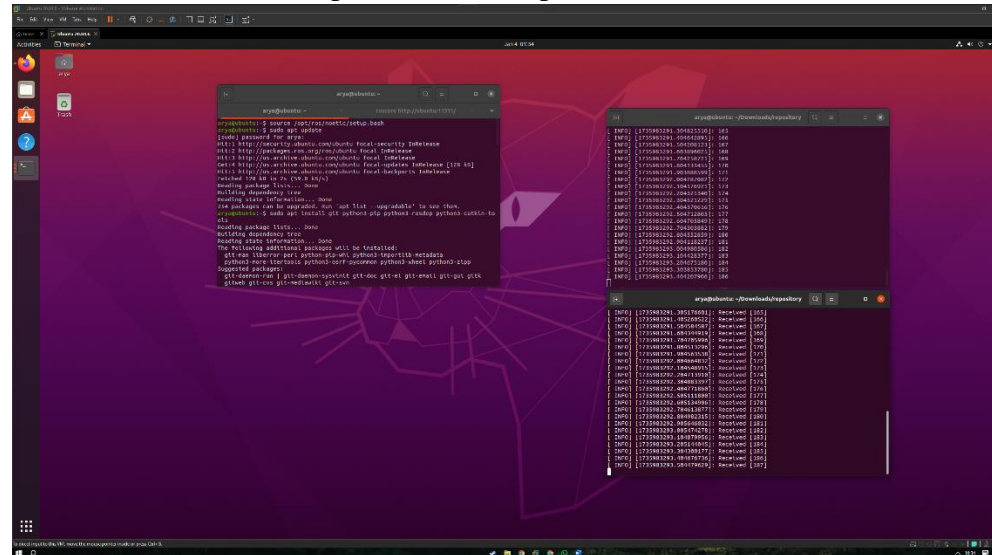
Keberhasilan instalasi diverifikasi dengan menjalankan perintah roscore, yang memastikan layanan inti ROS seperti ROS Master dan Parameter Server berfungsi dengan baik. Langkah-langkah ini memberikan dasar yang kuat untuk memulai pengembangan robotika menggunakan ROS.

Dokumentasi Chapter 2

1. `source /opt/ros/noetic/setup.bash` (Agar terminal Mengenali perintah dan paket ROS)
2. `sudo apt update` (agar daftar paket di sistem diperbarui)
3. Jika terjadi **error** pada step 2 maka lakukan : `sudo apt clean` dan ulangi step 2 sampai tidak ada **error**
4. `sudo apt install git python3-pip python3-rosdep python3-catkin-tools`
5. `sudo rosdep init`
6. Jika terjadi **error** pada step 5 maka `sudo rm /etc/ros/rosdep/sources.list.d/20-default.list` dan ulangi step 5 sampai tidak ada **error**
7. `rosdep update`
8. buka terminal baru dan ketik `"cd ~/Downloads"` untuk berpindah direktori file
9. lakukan cloning github yang dituju dengan `git clone https://github.com/PacktPublishing/Mastering-ROS-for-Robotics-Programming-Third-edition.git` repository
10. Pastikan dengan mengecek menggunakan `"ls"`
11. Membuat folder baru yaitu `"mkdir src"`
12. Memindahkan direktori chapter 2 ke src yang dibuat tadi dengan `"mv Chapter2 src/"`
13. Menginstall dependensi yang diperlukan oleh paket di folder src `"rosdep install --from-paths src --ignore-src -r -y"`
14. Membuat workspace ROS menggunakan catkin `"catkin_make"`
15. Memuat konfigurasi workspace ROS ke dalam terminal yang digunakan saat ini `"source devel/setup.bash"`
16. Menjalankan layanan inti ROS `"roscore"`
17. Membuka terminal baru harus menjalankan `"source ~/Downloads/repository/devel/setup.bash"` agar tidak terjadi error agar terminal mengenali workspace yang dibuat

NODE DEMO:

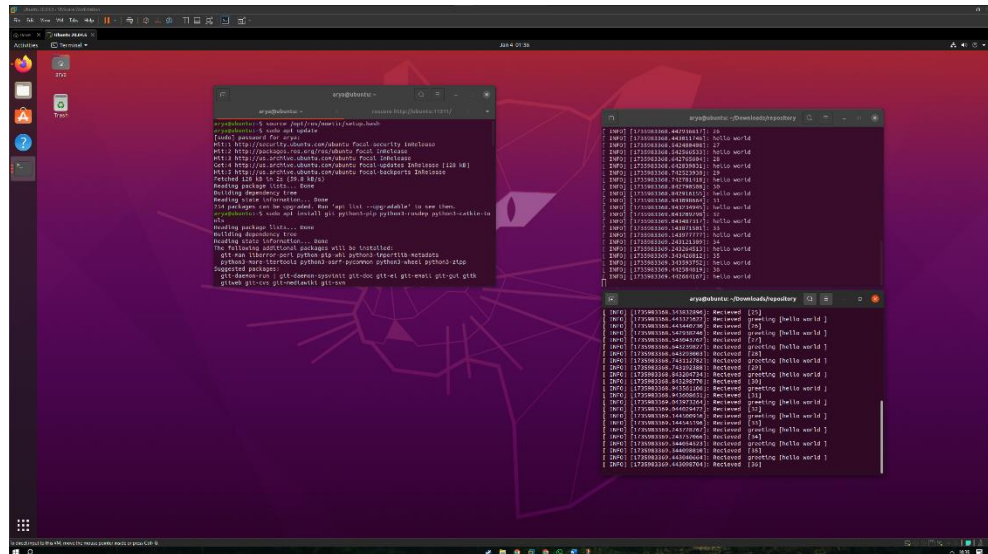
- `roslaunch masterling_ros_demo_pkg demo_topic_publisher` (menjalankan node publisher dari paket (`masterling_ros_demo_pkg`))
- `roslaunch masterling_ros_demo_pkg demo_topic_subscriber` (menjalankan node subscriber untuk menerima pesan dari node publisher)



Gambar 2.1 Node 1 berhasil dijalankan

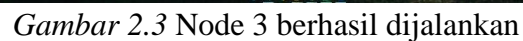
NODE MSG DEMO:

- `roslaunch masterling_ros_demo_pkg demo_msg_publisher`
- `roslaunch masterling_ros_demo_pkg demo_msg_subscriber`



Gambar 2.2 Node 2 berhasil dijalankan

- `roslaunch masterling_ros_demo_pkg demo_service_server`
- `roslaunch masterling_ros_demo_pkg demo_service_client`



Analisis Chapter 2

Dokumentasi Chapter 2 menjelaskan langkah-langkah untuk mengatur lingkungan kerja ROS dan menjalankan beberapa node demo dari paket *mastering_ros_demo_pkg*. Proses dimulai dengan memastikan terminal mengenali perintah dan paket ROS melalui perintah `source /opt/ros/noetic/setup.bash`, diikuti dengan pembaruan daftar paket menggunakan `sudo apt update`. Jika terjadi error, langkah tambahan seperti membersihkan cache dengan `sudo apt clean` dilakukan. Paket-paket pendukung seperti `git`, `python3-pip`, `python3-rosdep`, dan `python3-catkin-tools` diinstal untuk mendukung manajemen dependensi dan pembangunan proyek ROS.

Dependensi ROS disiapkan dengan `rosdep init` dan `rosdep update`, di mana error pada langkah ini dapat diatasi dengan menghapus file konfigurasi rusak menggunakan `sudo rm`. Setelah repositori proyek di-*clone* dari GitHub, struktur workspace disiapkan dengan membuat folder `src` dan memindahkan direktori Chapter2 ke dalamnya. Dependensi paket diinstal dengan `rosdep install`, workspace ROS dibangun menggunakan `catkin_make`, dan konfigurasi dimuat dengan `source devel/setup.bash`. Langkah ini diakhiri dengan menjalankan `roscore` untuk memastikan layanan inti ROS aktif.

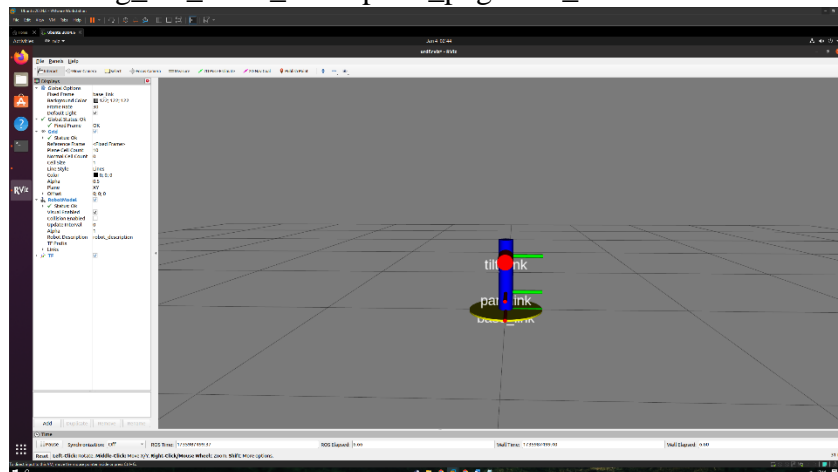
Selanjutnya, node demo dari paket *mastering_ros_demo_pkg* dijalankan untuk mempraktikkan komunikasi antar-node dan layanan ROS. Node pertama adalah `demo_topic_publisher` dan `demo_topic_subscriber`, yang menunjukkan bagaimana data dikirimkan melalui topik. Node berikutnya adalah `demo_msg_publisher` dan `demo_msg_subscriber`, yang menambahkan elemen pesan khusus dalam komunikasi. Terakhir, node `demo_service_server` dan `demo_service_client` memperlihatkan komunikasi berbasis layanan antara server dan klien. Setiap langkah diakhiri dengan verifikasi melalui gambar atau hasil keluaran terminal untuk memastikan setiap node berhasil dijalankan. Dokumentasi ini memberikan panduan yang sistematis untuk memahami dasar-dasar pengaturan dan implementasi komunikasi ROS menggunakan node, topik, pesan, dan layanan.

Dokumentasi Chapter 3

1. Membuat terminal mengenali perintah dan paket ros menggunakan “source /opt/ros/noetic/setup.bash”
2. cd ~/Downloads (Berpindah direktori ke Download)
3. cd ~/Downloads/repository (Berpindah direktori ke repository)
4. Memastikan file yang tersedia di dalam repository menggunakan “ls”
5. Membuat file src untuk kode sumber proyek “mkdir src” (Tidak usah membuat lg jika sudah ada sebelumnya)
6. Memindahkan folder chapter 3 ke dalam src “mv Chapter3/ src/” (Pastikan nama harus sesuai dan memilih ingin dipindahkan ke folder mana)
7. cd ~/Downloads/repository (Berpindah direktori kembali ke repository)
8. Jalankan “roscore” jika sebelumnya menutup atau belum pernah menjalankan sama sekali
9. Membuat workspace catkin dengan “catkin_make”
10. Memuat konfigurasi workspace ROS ke dalam terminal yang digunakan saat ini “source devel/setup.bash”
11. Menjalankan code untuk 3D Modeling Robot dengan aplikasi rviz

Simulasi 1:

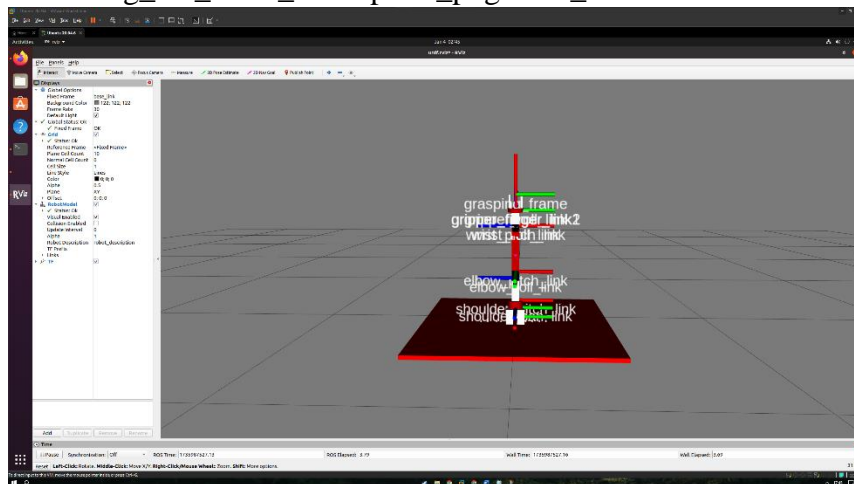
roslaunch mastering_ros_robot_description_pkg view_demo.launch



Gambar 3.1 Simulasi 3D Modeling 1

Simulasi 2:

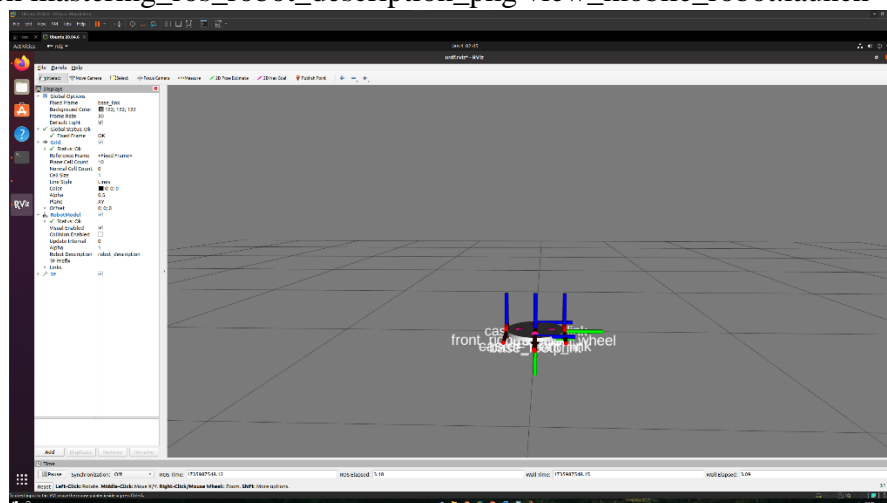
roslaunch mastering_ros_robot_description_pkg view_arm.launch



Gambar 3.2 Simulasi 3D Modeling 2

Simulasi 3:

roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch



Gambar 3.3 Simulasi 3D Modeling 3

Analisis Chapter 3

Pada video ketiga dalam playlist robotika Anda, tutorial akan membahas tentang Working with ROS for 3D Modeling yang diambil dari bab ketiga buku *Mastering ROS*. Video ini akan diawali dengan pengenalan diri dan tujuan dari tutorial, yaitu untuk memandu penonton memahami langkah-langkah penggunaan ROS dalam membuat model 3D. Proses dimulai dengan mengatur lingkungan ROS menggunakan perintah `source /opt/ros/noetic/setup.bash`, yang bertujuan agar terminal dapat mengenali perintah dan paket-paket ROS.

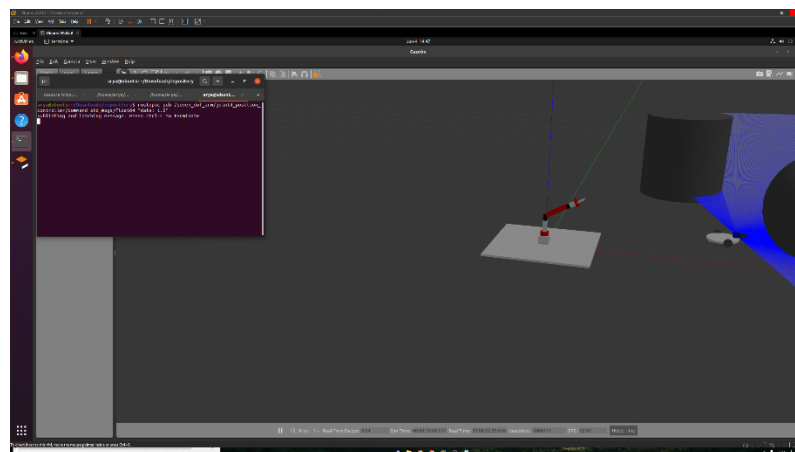
Selanjutnya, penonton akan diarahkan untuk berpindah direktori ke folder *Downloads/repository* dan membuat folder *src* untuk menyimpan kode sumber proyek. Setelah itu, folder *Chapter 3* dipindahkan ke dalam folder *src*, dilanjutkan dengan menjalankan perintah `roscore` jika belum dijalankan sebelumnya.

Langkah selanjutnya adalah membangun workspace dengan `catkin_make` untuk menyiapkan lingkungan kerja, lalu menjalankan `source devel/setup.bash` untuk mengenali paket-paket yang telah dibangun.

Kemudian, penonton akan diajarkan untuk meluncurkan berbagai file launch, yaitu `view_demo.launch`, `view_arm.launch`, dan `view_mobile_robot.launch`, yang masing-masing akan menampilkan tampilan 3D dari berbagai model robot yang berbeda. Setiap langkah dijelaskan secara rinci untuk memastikan pemahaman yang baik dalam bekerja dengan ROS untuk pemodelan 3D, sekaligus memberikan penonton alat dan teknik yang dibutuhkan untuk menerapkannya dalam proyek robotik mereka sendiri.

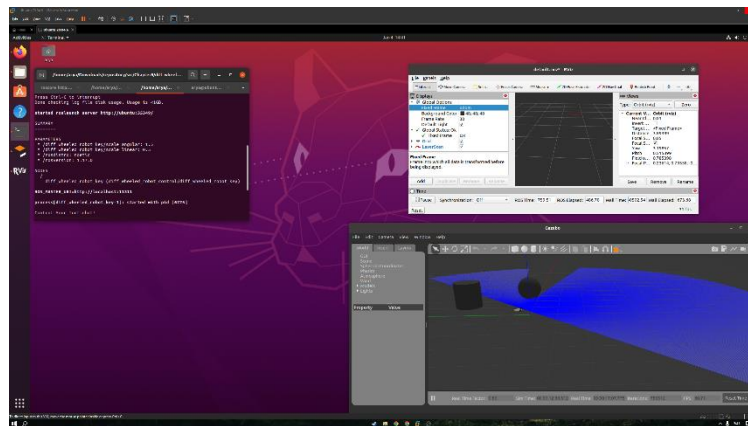
Dokumentasi Chapter 4

1. Berpindah direktori ke download “Step 1: `cd ~/Downloads`”
2. Berpindah direktori ke repository “`cd ~/Downloads/repository`”
3. Memastikan file apa saja didalam folder dengan “`ls`”
4. Memindahkan dari folder chapter 4 di repository ke dalam src dengan ‘`mv Chapter4/ src/`’
5. Memindahkan dari folder chapter 6 di repository ke dalam src dengan ‘`mv Chapter6/ src/`’
6. Jalankan “`roscore`”
7. Membuka tab terminal baru dan membuat catkin workspace “`catkin_make`”
8. Jangan lupa mengetik “`source devel/setup.bash`” untuk mengatur workspace agar dapat mengenali paket
9. Menjalan simulasi menggerakan joint robot dengan ROS di dalam Gazebo dengan mengetik “`roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch`”
10. Membuka Terminal baru dan mengetik “`rostopic pub /seven_dof_arm/joint4_position_controller/command std_msgs/Float64 "data: 1.0"`”



Gambar 4.1 Menggerakan Joint Robot dengan ROS

11. Melakukan simulasi Teleop Node menggunakan Gazebo dan mengontrol dengan turtlebot “`roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch`”
12. Dan ini code untuk controller nya “`roslaunch diff_wheeled_robot_control keyboard_teleop.launch`”
13. Buka terminal baru dan ketik “`rviz`”
14. Pilih odom pada bagian fixed frame
15. Pilih opsi *add* dan pilih *By topic* dan pilih scan di paling bawah yang bernama *Laser Scan*
16. Lalu jangan lupa berikan object agar camera dapat mendeteksi object pada rviz



Gambar 4.2 Menggerakan Turtlebot yang dapat mendeteksi objek

Analisis Chapter 4

Chapter 4 menjelaskan langkah-langkah dasar untuk mengatur, menjalankan, dan mengontrol simulasi robot menggunakan ROS dan Gazebo. Proses dimulai dengan pengelolaan struktur direktori, di mana folder yang relevan, seperti Chapter4 dan Chapter6, dipindahkan ke dalam direktori src di workspace ROS. Ini menunjukkan pentingnya pengorganisasian workspace agar ROS dapat mengenali semua paket yang dibutuhkan. Selanjutnya, proses pembangunan workspace menggunakan `catkin_make` dan penyiapan environment dengan `source devel/setup.bash` menjadi langkah kunci untuk memastikan workspace berfungsi secara optimal. Langkah-langkah ini menekankan bagaimana infrastruktur ROS mengintegrasikan berbagai komponen untuk simulasi yang mulus.

Selain itu, chapter ini memperkenalkan cara menggerakkan joint robot di Gazebo menggunakan perintah `rostopic pub`, yang memperlihatkan fleksibilitas ROS dalam mengontrol robot melalui komunikasi berbasis pesan. Integrasi dengan RViz memungkinkan visualisasi data sensor, seperti Laser Scan, untuk mendeteksi objek di lingkungan simulasi. Di sisi lain, simulasi kontrol menggunakan node teleoperation (`keyboard_teleop.launch`) menawarkan pendekatan interaktif untuk mengendalikan robot dalam simulasi. Pendekatan ini memperlihatkan bagaimana ROS dapat digunakan untuk menciptakan skenario simulasi realistis yang mendukung pengembangan dan pengujian algoritma kontrol robot. Secara keseluruhan, chapter ini memberikan gambaran komprehensif tentang bagaimana ROS, Gazebo, dan RViz saling melengkapi dalam menciptakan platform simulasi robotik.