

TECHNICAL REPORT UAS ROBOTIKA
Pytorch For Deep Learning

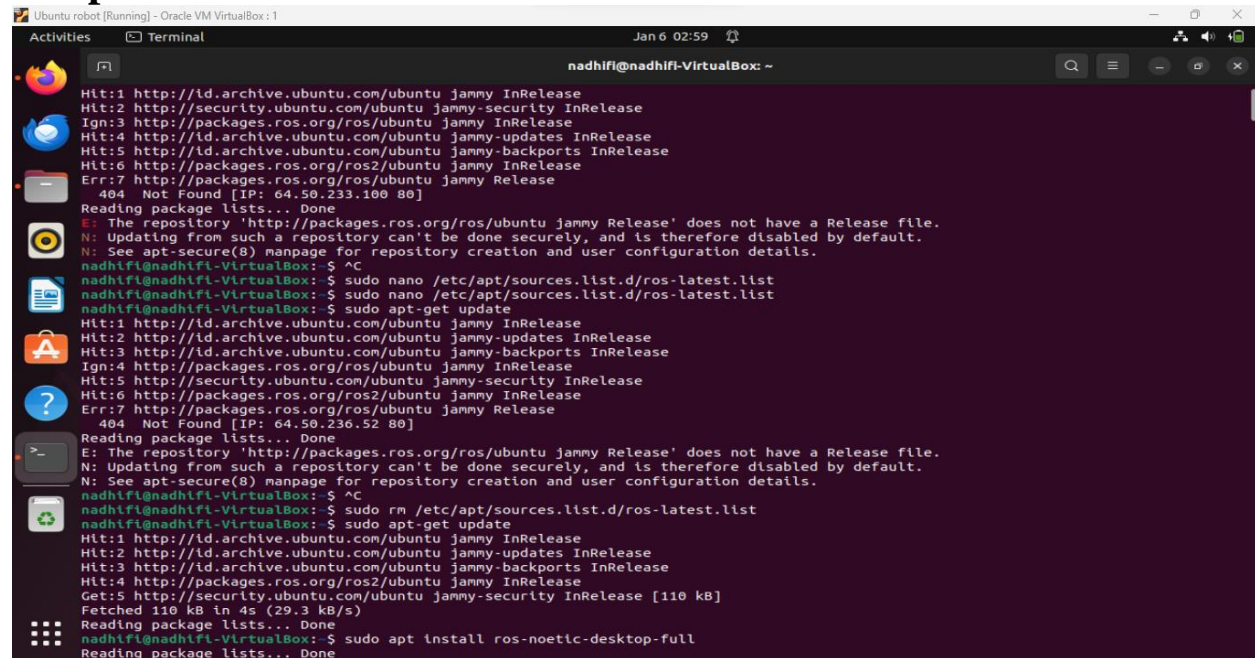


Oleh :

Nadhifi Qurrunul Bahratu Fauzan Hibatullah /1103204156

**PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG
2022**

Chapter 1



```
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:3 http://packages.ros.org/ros/ubuntu jammy InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://packages.ros.org/ros2/ubuntu jammy InRelease
Err:7 http://packages.ros.org/ros/ubuntu jammy Release
 404 Not Found [IP: 64.50.233.100 80]
Reading package lists... Done
E: The repository 'http://packages.ros.org/ros/ubuntu jammy Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
nadhifi@nadhifi-VirtualBox:~$ ^C
nadhifi@nadhifi-VirtualBox:~$ sudo nano /etc/apt/sources.list.d/ros-latest.list
nadhifi@nadhifi-VirtualBox:~$ sudo apt-get update
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://packages.ros.org/ros/ubuntu jammy InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:6 http://packages.ros.org/ros2/ubuntu jammy InRelease
Err:7 http://packages.ros.org/ros/ubuntu jammy Release
 404 Not Found [IP: 64.50.236.52 80]
Reading package lists... Done
E: The repository 'http://packages.ros.org/ros/ubuntu jammy Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
nadhifi@nadhifi-VirtualBox:~$ ^C
nadhifi@nadhifi-VirtualBox:~$ sudo rm /etc/apt/sources.list.d/ros-latest.list
nadhifi@nadhifi-VirtualBox:~$ sudo apt-get update
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://packages.ros.org/ros2/ubuntu jammy InRelease
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 110 kB in 4s (29.3 kB/s)
Reading package lists... Done
nadhifi@nadhifi-VirtualBox:~$ sudo apt install ros-noetic-desktop-full
Reading package lists... Done
```

Instalasi ROS Noetic pada Ubuntu melibatkan beberapa langkah. Berikut adalah panduan langkah-demi-langkah berserta analisis:

Langkah 1: Perbarui Repositori dan Upgrade Paket

Analisis:Langkah ini memastikan bahwa paket sistem terkini telah diunduh dan diinstal.

Langkah 2: Instalasi Dependencies

Analisis:ROS Noetic memiliki beberapa dependensi yang perlu diinstal sebelum instalasi ROS itu sendiri. Paket ini mencakup alat-alat yang diperlukan untuk mengelola workspace dan proyek ROS.

Langkah 3: Inisialisasi rosdep

Analisis: Rosdep adalah sistem manajemen dependensi di ROS. Inisialisasi dan pembaruan dibutuhkan untuk mendapatkan informasi dependensi terkini untuk paket ROS.

Langkah 4: Instalasi ROS Noetic Desktop-Full

Analisis: Instalasi paket `ros-noetic-desktop-full` mencakup instalasi ROS, tools dan library yang umum digunakan, serta beberapa simulator dan perangkat lunak pendukung.

Langkah 5: Konfigurasi Environment ROS

Analisis: Memastikan bahwa lingkungan ROS terkonfigurasi dengan baik setiap kali terminal dibuka.

Langkah 6: Membuat dan Inisialisasi Workspace

****Analisis:**** Workspace ROS dibuat dan diinisialisasi. Ini adalah tempat di mana paket ROS akan dikembangkan atau dikelola.

Langkah 7: Menjalankan roscore

Analisis: Menjalankan `roscore` memastikan bahwa master ROS sedang berjalan, yang diperlukan untuk berkomunikasi antara node-node ROS.

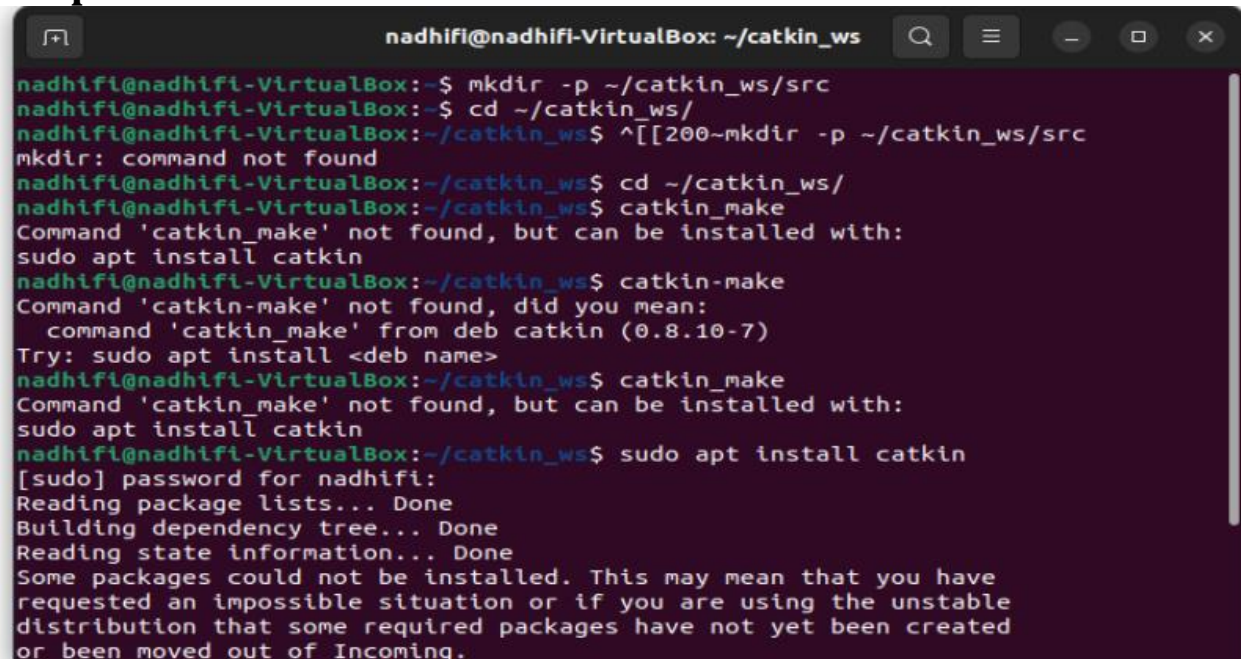
Langkah 8: Instalasi Paket Tambahan (Opsional)

Analisis: Jika diperlukan, paket ROS tambahan dapat diinstal untuk memenuhi kebutuhan spesifik proyek atau pengembangan.

Langkah 9: Verifikasi Instalasi

Analisis: Memeriksa apakah sistem dapat mengakses direktori ROS. Jika berhasil, instalasi ROS Noetic di Ubuntu telah berhasil.

Chapter 2



```
nadhifi@nadhifi-VirtualBox: ~/catkin_ws
nadhifi@nadhifi-VirtualBox:~$ mkdir -p ~/catkin_ws/src
nadhifi@nadhifi-VirtualBox:~$ cd ~/catkin_ws/
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ ^[[200~mkdir -p ~/catkin_ws/src
mkdir: command not found
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ cd ~/catkin_ws/
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ catkin_make
Command 'catkin_make' not found, but can be installed with:
sudo apt install catkin
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ catkin-make
Command 'catkin-make' not found, did you mean:
  command 'catkin_make' from deb catkin (0.8.10-7)
Try: sudo apt install <deb name>
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ catkin_make
Command 'catkin_make' not found, but can be installed with:
sudo apt install catkin
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ sudo apt install catkin
[sudo] password for nadhifi:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
```

ROS Lesson 02 - "catkin_make, catkin_create_pkg, create dependencies" membahas penggunaan beberapa perintah kunci dalam konteks pengembangan perangkat lunak ROS. Berikut adalah langkah-langkahnya:

1. Membuat Workspace: Jalankan perintah untuk membuat workspace baru dalam ROS.

Analisis: Pembuatan workspace ini penting untuk mengorganisir paket-paket ROS dan memfasilitasi pengembangan yang terstruktur.

2. Membuat Package: Gunakan perintah `catkin_create_pkg` untuk membuat paket ROS baru.

Analisis: Perintah ini membantu membuat struktur dasar paket, dan spesifikasi dependensi paket ditentukan di sini.

3. Mengaktifkan Workspace: Aktifkan workspace dengan menjalankan perintah berikut.

Analisis: Ini diperlukan untuk memastikan bahwa environment saat ini terkait dengan workspace yang sedang digunakan.

4. Menambahkan Dependencies: Gunakan perintah ``find_package`` untuk menemukan dan menambahkan dependencies ke dalam CMakeLists.txt di paket.

Analisis: Mendefinisikan dependensi yang dibutuhkan oleh paket, memungkinkan CMake untuk mengelola dan menyusun paket secara efisien.

5. Build Paket: Jalankan ``catkin_make`` untuk mengonstruksi atau mem-build paket-paket dalam workspace.

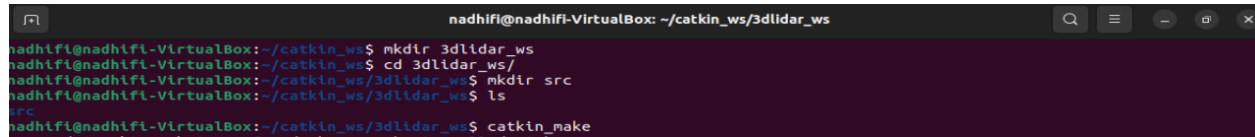
Analisis: Perintah ini menghasilkan executables dan libraries yang terkait dengan paket, memungkinkan eksekusi node-node ROS dan modul-modul lainnya.

6. Menjalankan ROS Node: Jalankan salah satu node yang ada di paket.

Analisis: Menjalankan node ini membuktikan bahwa proses pembuatan paket dan dependensinya berjalan dengan baik, dan node dapat beroperasi dengan benar.

Secara keseluruhan, Lesson 02 ini memberikan pemahaman yang baik tentang bagaimana membangun workspace, membuat paket, menambahkan dependencies, dan menjalankan node pada lingkungan ROS. Hal ini penting untuk pengembangan ROS yang efektif dan terstruktur serta membentuk dasar bagi pemahaman konsep lebih lanjut dalam pengembangan perangkat lunak robotik.

Chapter 3

A terminal window titled 'nadhifi@nadhifi-VirtualBox: ~/catkin_ws/3dlidar_ws' with a dark purple background. It shows a sequence of commands: 'mkdir 3dlidar_ws', 'cd 3dlidar_ws/', 'mkdir src', 'ls', and 'catkin_make'. The output shows the 'src' directory being created.

```
nadhifi@nadhifi-VirtualBox: ~/catkin_ws/3dlidar_ws
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ mkdir 3dlidar_ws
nadhifi@nadhifi-VirtualBox:~/catkin_ws$ cd 3dlidar_ws/
nadhifi@nadhifi-VirtualBox:~/catkin_ws/3dlidar_ws$ mkdir src
nadhifi@nadhifi-VirtualBox:~/catkin_ws/3dlidar_ws$ ls
src
nadhifi@nadhifi-VirtualBox:~/catkin_ws/3dlidar_ws$ catkin_make
```

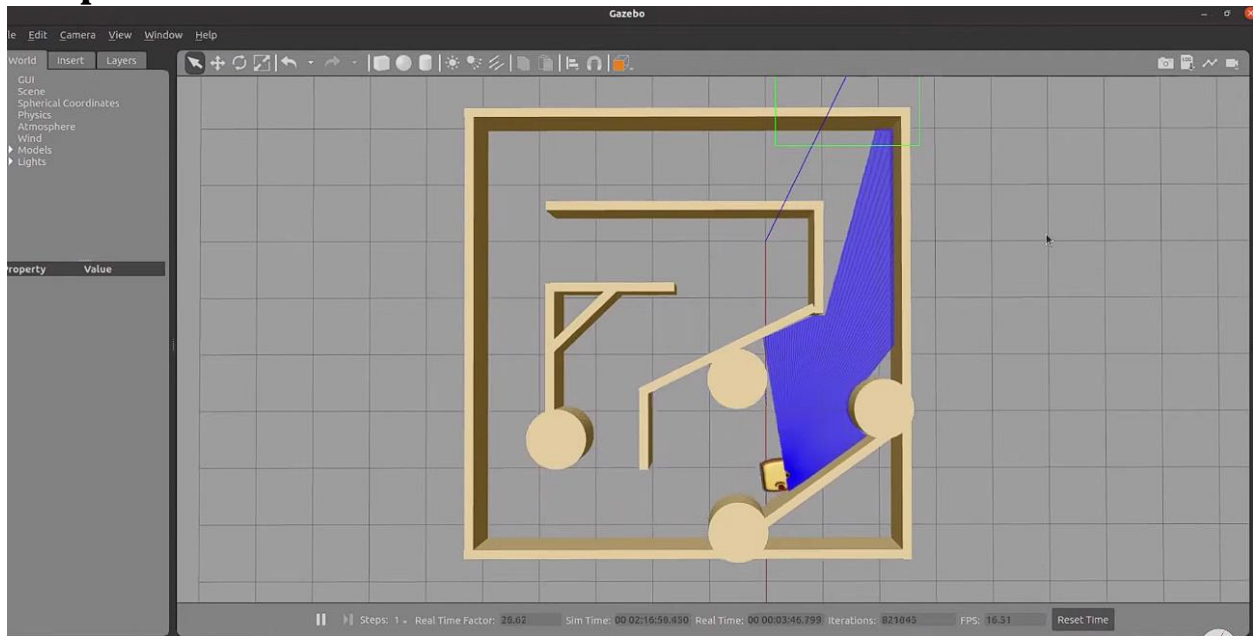
"Mastering ROS for Robotics Programming, Third Edition" menyajikan langkah-langkah mendalam untuk bekerja dengan ROS dalam konteks pemodelan 3D. Berikut langkah-langkah utama dan analisisnya:

1. Pemahaman Konsep ROS: Mulai dengan memahami konsep dasar ROS, termasuk topik-topik seperti node, topic, service, dan bagaimana mengintegrasikan komponen-komponen ini dalam sistem robotika.
2. Pemahaman Gazebo Simulator: Pelajari penggunaan simulator robotik Gazebo yang terintegrasi dengan ROS untuk mensimulasikan dan menguji model robot dalam lingkungan 3D.
3. Pemodelan Robot Menggunakan URDF: Gunakan Universal Robot Description Format (URDF) untuk memodelkan robot dan lingkungan dalam bentuk XML, mencakup penempatan sensor, aktuator, dan link.
4. Visualisasi dengan RViz: Pahami penggunaan RViz untuk visualisasi model robot dan analisis data sensor yang dihasilkan oleh robot dalam simulasi 3D.
5. Menggunakan PointCloud dan OctoMap: Pelajari cara bekerja dengan data PointCloud untuk merepresentasikan objek dan lingkungan dalam ruang tiga dimensi, serta bagaimana OctoMap digunakan untuk membangun peta oktagon dari lingkungan.
6. Navigasi dengan MoveIt!: Terapkan MoveIt! untuk perencanaan pergerakan dan navigasi robot dalam lingkungan 3D. Ini melibatkan konfigurasi dan pemrograman kontroler gerak untuk melakukan pergerakan yang optimal.

Analisis:

Buku ini memberikan pemahaman yang mendalam tentang penerapan ROS dalam konteks pemodelan 3D dan simulasi robotika. Dengan menyajikan langkah-langkah praktis dan contoh nyata, pembaca dapat memperoleh keterampilan yang kuat dalam mengelola dan mengembangkan aplikasi robotika yang kompleks. Materi yang disajikan mencakup berbagai aspek, mulai dari pemodelan robot hingga navigasi dalam lingkungan 3D, sehingga memberikan pandangan komprehensif dalam penggunaan ROS dalam pengembangan robotika.

Chapter 4



Langkah-langkah Simulasi Robot Menggunakan ROS dan Gazebo

1. Instalasi ROS dan Gazebo

Instalasi ROS dan Gazebo dapat dilakukan sesuai dengan petunjuk yang tersedia di situs web ROS.

2. Buat model robot

Model robot dapat dibuat menggunakan berbagai perangkat lunak CAD, seperti SolidWorks, Blender, atau Gazebo sendiri. Model robot yang dibuat harus menggunakan format file XACRO.

3. Impor model robot ke Gazebo

Model robot yang telah dibuat dapat diimpor ke Gazebo menggunakan perintah `gazebo --verbose`.

4. Aktifkan kontrol ROS

Untuk mengaktifkan kontrol ROS pada robot, perlu ditambahkan plugin `gazebo_ros_control` ke model robot.

5. Buat node teleop

Node teleop digunakan untuk mengontrol robot secara manual menggunakan keyboard atau joystick.

Analisis

Buku "Mastering ROS for Robotics Programming, Third edition" memberikan panduan yang komprehensif untuk simulasi robot menggunakan ROS dan Gazebo. Buku ini mencakup langkah-langkah dasar hingga lanjutan, mulai dari instalasi ROS dan Gazebo, pembuatan model robot, hingga kontrol robot.

Buku ini sangat bermanfaat bagi para praktisi dan peneliti robotika yang ingin mempelajari simulasi robot menggunakan ROS dan Gazebo. Buku ini juga dapat digunakan sebagai referensi bagi para mahasiswa yang mengambil mata kuliah robotika.

Berikut adalah beberapa poin penting dari buku ini:

- ROS adalah platform perangkat lunak yang dapat digunakan untuk mengembangkan aplikasi robotika secara umum.
- Gazebo adalah simulator robot 3D yang dapat digunakan untuk mensimulasikan perilaku robot di dunia nyata.
- Untuk mensimulasikan robot menggunakan ROS dan Gazebo, perlu dilakukan langkah-langkah berikut:
 - Instalasi ROS dan Gazebo
 - Buat model robot
 - Impor model robot ke Gazebo
 - Aktifkan kontrol ROS
 - Buat node teleop

1. ROS:

- Instalasi ROS: Ikuti panduan resmi untuk menginstal versi ROS yang sesuai pada sistem operasi Anda.
- Buat Workspace ROS: Buat workspace untuk menyimpan paket dan file proyek Anda.
- Buat atau pilih model robot: Gunakan model robot yang sudah ada atau buat model baru menggunakan URDF atau xacro.
- Tulis kontroler ROS: Buat node ROS untuk mengendalikan perilaku robot.
- Launch file: Buat file launch untuk memulai simulasi, termasuk model robot, kontroler, dan komponen ROS lainnya.
- Jalankan simulasi: Gunakan perintah `roslaunch` untuk menjalankan simulasi.

- Instalasi CoppeliaSim: Unduh dan instal CoppeliaSim dari situs resmi.
- Buka atau buat model robot: Buka model robot yang sudah ada atau buat model baru menggunakan antarmuka grafis CoppeliaSim.
- Tulis kontroler: Buat kontroler menggunakan bahasa pemrograman yang didukung CoppeliaSim (C/C++, Python, Lua, Java).
- Simulasi: Jalankan simulasi dengan menekan tombol "Run" atau menggunakan perintah dalam kontroler.

- Instalasi Webots: Unduh dan instal Webots dari situs resmi.
- Buka atau buat 'world': Buka 'world' yang sudah ada atau buat 'world' baru menggunakan antarmuka grafis Webots.
- Tulis kontroler: Buat kontroler menggunakan bahasa pemrograman yang didukung Webots (C/C++, Java, Python, MATLAB).
- Simulasi: Jalankan simulasi dengan menekan tombol "Run" atau menggunakan perintah dalam kontroler.

Analisis keseluruhan:

Pemilihan simulator yang tepat bergantung pada kebutuhan spesifik proyek. ROS menawarkan fleksibilitas dan integrasi dengan berbagai alat dan pustaka, CoppeliaSim unggul dalam visualisasi dan model fisika yang realistis, sedangkan Webots menyediakan fitur-fitur untuk pendidikan dan penelitian robotika.