Nama : Enrico Andreson Pattipeilohy

NIM : 1103204206

REPORT UTS

Pada kesempatan kali ini, saya akan menjelaskan apa saja yang saya lakukan dalam video ini https://youtu.be/g3bOTY7G9NY

SETUP

Untuk menjalakan ROS 2 atau ROS Melodic, kita harus menginstal ubuntu yang mensuport system tersebut. dalam hal ini ubuntu yang mensuport ROS 2 atau ROS Melodic adalah ubuntu versi 18.

SESI INSTALASI

1. Konfigurasi Repositori Ubuntu:

- Langkah ini memungkinkan akses ke lebih banyak paket perangkat lunak. Kita perlu mengikuti panduan terpisah yang disediakan oleh Ubuntu untuk detail tentang mengaktifkan repositori "restricted," "universe," dan "multiverse".

2. Menyiapkan sources.list:

- Ini menambahkan sumber baru untuk paket ROS ke daftar paket system, Perintah tersebut membuat file bernama `ros-latest.list` di direktori `/etc/apt/sources.list.d/`.

3. Menyiapkan Keys:

- Ini memastikan keaslian perangkat lunak yang diunduh dari sumber paket ROS.
    - Pertama, perintah memeriksa apakah `curl` terinstal (alat untuk mengunduh file). Jika tidak, maka akan diinstal.
    - Kemudian, perintah tersebut mengunduh kunci dari GitHub dan menambahkannya ke daftar kunci terpercaya sistem.

4. Memperbarui Indeks Paket:

- Ini memperbarui daftar paket yang tersedia dari semua sumber di sistem, termasuk repositori ROS yang baru ditambahkan.

5. Memilih Jenis Instalasi ROS:

- ROS menawarkan berbagai opsi instalasi tergantung kebutuhan:
    - **Desktop-Full Install (Disarankan):** Mencakup semua fungsi inti ROS, alat antarmuka grafis (`rqt`, `rviz`), pustaka robot, simulator, dan paket persepsi.
    - **Desktop Install:** Menyediakan fungsi inti ROS, alat antarmuka grafis (`rqt`, `rviz`), dan pustaka robot.
    - **ROS-Base (Minimal):** Hanya menginstal pustaka ROS penting untuk membangun dan komunikasi, tanpa alat grafis.
    - **Paket Individual:** Memungkinkan Anda memasang paket ROS tertentu dengan mengganti garis bawah di nama paket dengan tanda hubung (misalnya, `ros-melodic-slam-gmapping`). Gunakan `apt search ros-melodic` untuk menemukan paket yang tersedia.

6. Menyiapkan Lingkungan ROS:

- Ini mengkonfigurasi terminal Anda untuk mengenali perintah ROS. Perintah yang diberikan menambahkan baris ke file konfigurasi shell Anda (`.bashrc` untuk bash atau `.zshrc` untuk zsh) yang secara otomatis menyiapkan lingkungan ROS setiap kali Anda membuka jendela terminal baru.

7. Memasang Dependensi untuk Membangun Paket:

- Langkah sebelumnya memungkinkan Anda untuk menjalankan paket ROS yang ada, langkah ini menginstal alat tambahan yang diperlukan untuk membuat dan mengelola ruang kerja ROS sendiri. Alat-alat ini termasuk:
    - `python-rosdep`: Alat manajemen dependensi untuk paket ROS.
    - `python-rosinstall`, `python-rosinstall-generator`, `python-wstool`: Alat untuk mengunduh dan mengelola kode sumber ROS.
    - `build-essential`: Alat pengembangan yang diperlukan untuk membangun paket ROS dari sumber.

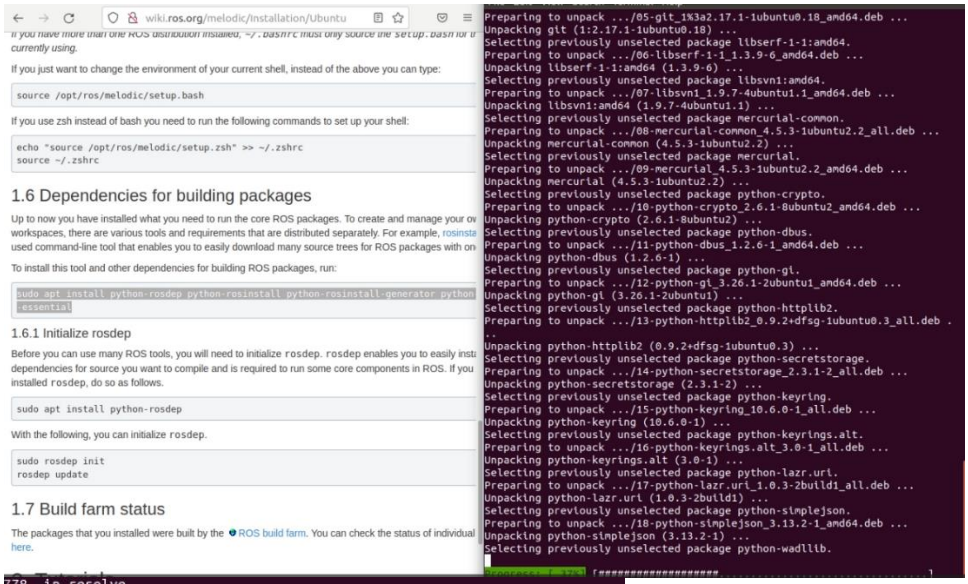
8. Inisialisasi rosdep:

- `rosdep` membantu mengelola dependensi untuk paket ROS. Perintah ini menginstal `rosdep` (jika belum terinstal) dan menginisialisasinya.

Dengan mengikuti langkah-langkah ini, ROS akan terinstal dan siap digunakan di sistem Ubuntu.

```
arch@march-VirtualBox:~$ sudo sh -c 'echo "deb http://packages.ros.org/ro
s/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.lis
t'
[sudo] password for march:
arch@march-VirtualBox:~$ sudo sh -c 'echo "deb http://packages.ros.org/ro
s/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.lis
t'
arch@march-VirtualBox:~$ sudo apt install curl # if you haven't already i
nstalled curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.24).
0 upgraded, 0 newly installed, 0 to remove and 267 not upgraded.
arch@march-VirtualBox:~$ curl -s https://raw.githubusercontent.com/ros/ro
sdistro/master/ros.asc | sudo apt-key add -
OK
arch@march-VirtualBox:~$ sudo apt update
Hit:1 http://packages.ros.org/ros/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
267 packages can be upgraded. Run 'apt list --upgradable' to see them.
arch@march-VirtualBox:~$ sudo apt install ros-melodic-desktop-full
Reading package lists... Done
Building dependency tree
Reading state information... Done
ros-melodic-desktop-full is already the newest version (1.4.1-0bionic.2023
0620.175308).
0 upgraded, 0 newly installed, 0 to remove and 267 not upgraded.
arch@march-VirtualBox:~$ echo "source /opt/ros/melodic/setup.bash" >> ~/.
bashrc
arch@march-VirtualBox:~$
```

```
o all 0.9.0-100 [6.388 B]
Get:27 http://packages.ros.org/ros/ubuntu bionic/main amd64 python-vcstool
s all 0.1.42-1 [30,1 kB]
Get:28 http://packages.ros.org/ros/ubuntu bionic/main amd64 python-wstool
ll 0.1.17-1 [42,4 kB]
Get:29 http://packages.ros.org/ros/ubuntu bionic/main amd64 python-rosinst
ll all 0.7.8-1 [23,7 kB]
Get:30 http://packages.ros.org/ros/ubuntu bionic/main amd64 python-rosinst
ll-generator all 0.1.23-1 [11,5 kB]
Fetched 11,8 MB in 3s (4.436 kB/s)
Selecting previously unselected package python-configobj.
(Reading database ... 246327 files and directories currently installed.)
Preparing to unpack .../00-python-configobj_5.0.6-2_all.deb ...
Unpacking python-configobj (5.0.6-2) ...
Selecting previously unselected package python-bzrlib.
Preparing to unpack .../01-python-bzrlib_2.7.0+bzr6622-10_amd64.deb ...
Unpacking python-bzrlib (2.7.0+bzr6622-10) ...
Selecting previously unselected package bzr.
Preparing to unpack .../02-bzr_2.7.0+bzr6622-10_all.deb ...
Unpacking bzr (2.7.0+bzr6622-10) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../03-liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../04-git-man_1%3a2.17.1-1ubuntu0.18_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.18) ...
Selecting previously unselected package git.
Preparing to unpack .../05-git_1%3a2.17.1-1ubuntu0.18_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.18) ...
Selecting previously unselected package libserf-1-1:amd64.
Preparing to unpack .../06-libserf-1-1_1.3.9-6_amd64.deb ...
Unpacking libserf-1-1:amd64 (1.3.9-6) ...
Selecting previously unselected package libsvn1:amd64.
Preparing to unpack .../07-libsvn1_1.9.7-4ubuntu1.1_amd64.deb ...
Unpacking libsvn1:amd64 (1.9.7-4ubuntu1.1) ...
Selecting previously unselected package mercurial-common.
Preparing to unpack .../08-mercurial-common_4.5.3-1ubuntu2.2_all.deb ...
Unpacking mercurial-common (4.5.3-1ubuntu2.2) ...
Selecting previously unselected package mercurial.
Preparing to unpack .../09-mercurial_4.5.3-1ubuntu2.2_amd64.deb ...
Unpacking mercurial (4.5.3-1ubuntu2.2) ...
Selecting previously unselected package python-crypto.
Preparing to unpack .../10-python-crypto_2.6.1-8ubuntu2_amd64.deb ...
```



wiki.ros.org/melodic/Installation/Ubuntu

If you have more than one ROS distribution installed, ~/.bashrc must only source the `setup.bash` for the version you are currently using.

If you just want to change the environment of your current shell, instead of the above you can type:

```
source /opt/ros/melodic/setup.bash
```

If you use zsh instead of bash you need to run the following commands to set up your shell:

```
echo "source /opt/ros/melodic/setup.zsh" >> ~/.zshrc
source ~/.zshrc
```

1.6 Dependencies for building packages

Up to now you have installed what you need to run the core ROS packages. To create and manage your ov
workspaces, there are various tools and requirements that are distributed separately. For example, rosinsta
used command-line tool that enables you to easily download many source trees for ROS packages with on

To install this tool and other dependencies for building ROS packages, run:

```
sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python
-essential
```

1.6.1 Initialize rosdep

Before you can use many ROS tools, you will need to initialize `rosdep`. `rosdep` enables you to easily insta
dependencies for source you want to compile and is required to run some core components in ROS. If you
installed `rosdep`, do so as follows.

```
sudo apt install python-rosdep
```

With the following, you can initialize `rosdep`.

```
sudo rosdep init
rosdep update
```

1.7 Build farm status

The packages that you installed were built by the ROS build farm. You can check the status of individual
here.

```
Preparing to unpack .../05-git_1%3a2.17.1-1ubuntu0.18_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.18) ...
Selecting previously unselected package libserf-1-1:amd64.
Preparing to unpack .../06-libserf-1-1_1.3.9-6_amd64.deb ...
Unpacking libserf-1-1:amd64 (1.3.9-6) ...
Selecting previously unselected package libsvn1:amd64.
Preparing to unpack .../07-libsvn1_1.9.7-4ubuntu1.1_amd64.deb ...
Unpacking libsvn1:amd64 (1.9.7-4ubuntu1.1) ...
Selecting previously unselected package mercurial-common.
Preparing to unpack .../08-mercurial-common_4.5.3-1ubuntu2.2_all.deb ...
Unpacking mercurial-common (4.5.3-1ubuntu2.2) ...
Selecting previously unselected package mercurial.
Preparing to unpack .../09-mercurial_4.5.3-1ubuntu2.2_amd64.deb ...
Unpacking mercurial (4.5.3-1ubuntu2.2) ...
Selecting previously unselected package python-crypto.
Preparing to unpack .../10-python-crypto_2.6.1-8ubuntu2_amd64.deb ...
Unpacking python-crypto (2.6.1-8ubuntu2) ...
Selecting previously unselected package python-dbus.
Preparing to unpack .../11-python-dbus_1.2.6-1_amd64.deb ...
Unpacking python-dbus (1.2.6-1) ...
Selecting previously unselected package python-gi.
Preparing to unpack .../12-python-gi_3.26.1-2ubuntu1_amd64.deb ...
Unpacking python-gi (3.26.1-2ubuntu1) ...
Selecting previously unselected package python-httplib2.
Preparing to unpack .../13-python-httplib2_0.9.2+dfsg-1ubuntu0.3_all.deb .
..
Unpacking python-httplib2 (0.9.2+dfsg-1ubuntu0.3) ...
Selecting previously unselected package python-secretstorage.
Preparing to unpack .../14-python-secretstorage_2.3.1-2_all.deb ...
Unpacking python-secretstorage (2.3.1-2) ...
Selecting previously unselected package python-keyring.
Preparing to unpack .../15-python-keyring_10.6.0-1_all.deb ...
Unpacking python-keyring (10.6.0-1) ...
Selecting previously unselected package python-keyrings.alt.
Preparing to unpack .../16-python-keyrings.alt_3.0-1_all.deb ...
Unpacking python-keyrings.alt (3.0-1) ...
Selecting previously unselected package python-lazr.uri.
Preparing to unpack .../17-python-lazr.uri_1.0.3-2build1_all.deb ...
Unpacking python-lazr.uri (1.0.3-2build1) ...
Selecting previously unselected package python-simplejson.
Preparing to unpack .../18-python-simplejson_3.13.2-1_amd64.deb ...
Unpacking python-simplejson (3.13.2-1) ...
Selecting previously unselected package python-wadllib.
```

```
778, in resolve
    raise DistributionNotFound(req, requirers)
pkg_resources.DistributionNotFound: The 'importlib_metadata' distribution
was not found and is required by rosdep
march@march-VirtualBox:~$ mkdir -p -/mlc/src
mkdir: invalid option -- '/'
Try 'mkdir --help' for more information.
march@march-VirtualBox:~$ mkdir -p ~/mymlc/src
march@march-VirtualBox:~$ cd ~/mymlc/
march@march-VirtualBox:~/mymlc$ catkin_make
Base path: /home/march/mymlc
Source space: /home/march/mymlc/src
Build space: /home/march/mymlc/build
Devel space: /home/march/mymlc/devel
Install space: /home/march/mymlc/install
Creating symlink "/home/march/mymlc/src/CMakeLists.txt" pointing to "/opt/
ros/melodic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/march/mymlc/src -DCATKIN_DEVEL_PREFIX=/
home/march/mymlc/devel -DCMAKE_INSTALL_PREFIX=/home/march/mymlc/install -G
 Unix Makefiles" in "/home/march/mymlc/build"
####
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/march/mymlc/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/melodic
-- This workspace overlays: /opt/ros/melodic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.17",
minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
```