

Nama : Faraday Barr Fatahillah
NIM : 1103213028
Kelas : TK-45-02

ROBOTICS FINALS

BAB 1

Pada bab ini, kita mempelajari perkenalan dari ROS seperti alasan menggunakan ROS sebagai sistem operasi utama dari robot, level dari file sistem ROS, grafis level komputasi, dan level komunitas ROS.

Alasan penggunaan utama menggunakan ROS adalah karena ROS sudah sangat terbaru dan paling berpotensi dalam jenisnya. Selain itu, terdapat banyak sekali alat dan fungsi untuk mempermudah visualisasi, coding, dan simulasi. Kemudian, ROS juga sangat support untuk sensor kelas atas.

Kemudian ROS juga memiliki level file sistem yang berbeda-beda dengan metapackages, packages, dan dengan urutan yang setara package manifest, messages, services, codes, dan misc. Packages adalah elemen sentral untuk program ROS karena untuk membuat paket ROS. Kemudian Package Manifest untuk segala hal berisi package, author, license, dependencies, compilation flags, dan banyak lagi. Selain itu ada metapackages yang biasanya berisi paket virtual dan tidak berisi kode atau file seperti biasannya.

Ada beberapa komputasi graf seperti Nodes, master, parameter server, messages, topics, services, dan bags. Nodes adalah proses yang memiliki komputasi. Master memberikan nama registrasi dan lookup processes untuk nodes yang lain. Parameter server digunakan untuk memasukkan data ke lokasi sentral. Topics adalah setiap message di ROS yang dikirimkan dengan bus.

BAB 2

Ubuntu 20.04 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Jun 12, 19:40

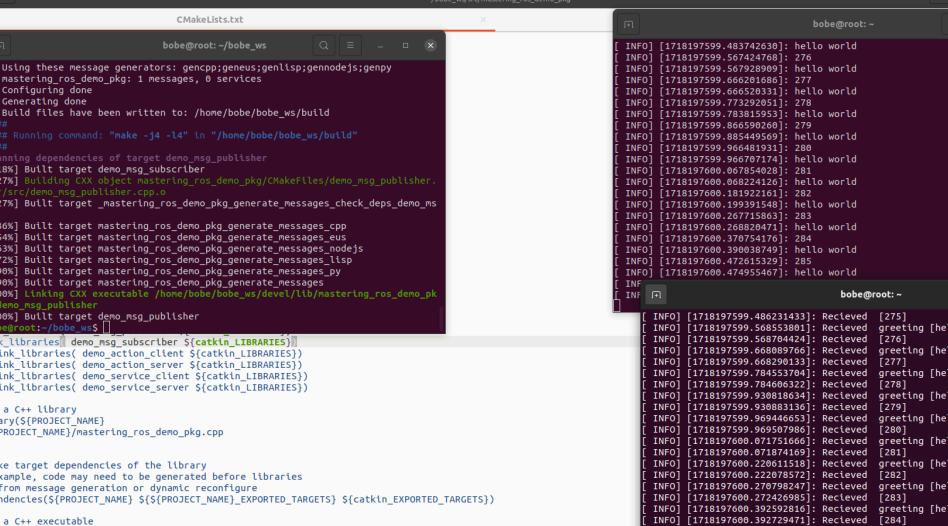
Recent Stared Home bobe_ws src mastering_demo_pkg

bobe@root:~/\$ catkin 0.8.10
...
BUILD_SHARED_LIBS is on
BUILD_SHARED_LIBS is on
...
traversing 1 packages in topological order:
- mastering_ros_demo_pkg
...
processing catkin package: 'mastering_ros_demo_pkg'
add_subdirectory(mastering_ros_demo_pkg)
Configuring done
Generating done
Building files have been written to: /home/bobe/bobe_ws/build
Running command: "make -j4 -l4" in "/home/bobe/bobe_ws/build"
Scanning dependencies of target demo_topic_publisher
[25%] Building CXX object mastering_ros_demo_pkg/CMakeFiles/demo_topic_publisher.dir/src/demo_topic_publisher.cpp.o
[75%] Built target demo_topic_subscriber
[100%] Built target demo_topic_publisher
[100%] Built target demo_topic_publisher
bobe@root:~/\$

bobe@root:~/\$ [1718196055, 913249541]: 158
[INFO] [1718196056, 013735931]: 159
[INFO] [1718196056, 014732231]: 160
[INFO] [1718196056, 214132544]: 161
[INFO] [1718196056, 317061129]: 162
[INFO] [1718196056, 413175111]: 163
[INFO] [1718196056, 513241526]: 164
[INFO] [1718196056, 613341545]: 165
[INFO] [1718196056, 713115129]: 166
[INFO] [1718196056, 815703373]: 167
[INFO] [1718196056, 912918902]: 168
[INFO] [1718196057, 015244134]: 169
[INFO] [1718196057, 114623981]: 170
[INFO] [1718196057, 217187571]: 171
[INFO] [1718196057, 319226150]: 172
[INFO] [1718196057, 423359529]: 173
[INFO] [1718196057, 516339424]: 174
[INFO] [1718196057, 616439443]: 175
[INFO] [1718196057, 713600137]: 176
[INFO] [1718196057, 814039712]: 177
[INFO] [1718196057, 917429613]: 178
[INFO] [1718196058, 017725424]: 179
[INFO] [1718196058, 114089124]: 180

bobe@root:~/\$ [1718196055, 913825289]: Received [158]
[INFO] [1718196056, 014732231]: Received [159]
[INFO] [1718196056, 114623981]: Received [160]
[INFO] [1718196056, 215163882]: Received [161]
[INFO] [1718196056, 316263883]: Received [162]
[INFO] [1718196056, 415495943]: Received [163]
[INFO] [1718196056, 514413591]: Received [164]
[INFO] [1718196056, 615386532]: Received [165]
[INFO] [1718196056, 71137231]: Received [166]
[INFO] [1718196056, 816485666]: Received [167]
[INFO] [1718196056, 914373127]: Received [168]
[INFO] [1718196057, 016085666]: Received [169]
[INFO] [1718196057, 113610208]: Received [170]
[INFO] [1718196057, 218594842]: Received [171]
[INFO] [1718196057, 320795241]: Received [172]
[INFO] [1718196057, 424918761]: Received [173]
[INFO] [1718196057, 518830170]: Received [174]
[INFO] [1718196057, 618452755]: Received [175]
[INFO] [1718196057, 714459814]: Received [176]
[INFO] [1718196057, 815310639]: Received [177]
[INFO] [1718196057, 919415507]: Received [178]

Berikut merupakan setelah mengikuti bagian topic publisher dan subscriber



```
112 # CATEK
113 # DEPEN
114 #
115 #-- Using these message generators: genccpp;genlisp;genodejs;genpy
116 #
117 #-----configuring done
118 #-----Build files have been written to: /home/bobe/bobe_ws/build
119 #
120 ## Spec
121 ## Running command: "make -j4 <14" in "/home/bobe/bobe_ws/build"
122 ## Includes
123 #include<ros.h>
124 #include<ros/ros.h>
125 #include<ros/console.h>
126 #include<ros/package.h>
127 #include<ros/master.h>
128 #include<ros/init.h>
129 #include<ros/node_handle.h>
130 #include<ros/service.h>
131 #include<ros/service_traits.h>
132 #include<ros/service_traits.h>
133 #include<ros/service_traits.h>
134 #include<ros/service_traits.h>
135 #include<ros/service_traits.h>
136 #include<ros/service_traits.h>
137 #include<ros/service_traits.h>
138 target<name=bobe>/home/bobe/bobe_ws/build
139 target<name=mastering_ros_demo_pkgs_subscriber> ${catkin_LIBRARIES}
140 target<name=mastering_ros_demo_pkgs_publisher> ${catkin_LIBRARIES}
141 #target<name=mastering_ros_demo_pkgs_server> ${catkin_LIBRARIES}
142 #target<name=mastering_ros_demo_pkgs_client> ${catkin_LIBRARIES}
143 #target<name=mastering_ros_demo_pkgs_service> ${catkin_LIBRARIES}
144
145 # Declare a C++ library
146 # add_library(${PROJECT_NAME})
147 # src/S(${PROJECT_NAME})/mastering_ros_demo_pkgs.cpp
148 # )
149
150 ## Add cmake target dependencies of the library
151 ## as an example, code may need to be generated before libraries
152 ## either from message generation or dynamic reconfigure
153 # add_dependencies(${PROJECT_NAME} ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
154
155 ## Declare a C++ executable
156 ## With catkin_make all packages are built within a single CMake context
157 ## The recommended prefix ensures that target names across packages don't collide
158 # add_executable(${PROJECT_NAME}_node src/mastering_ros_demo_pkgs_node.cpp)
```

[INFO] [1718197599, 487342630]: hello world
[INFO] [1718197599, 506853801]: hello world
[INFO] [1718197599, 569738946]: hello world
[INFO] [1718197599, 666291086]: 277
[INFO] [1718197599, 666520331]: hello world
[INFO] [1718197599, 773292651]: 278
[INFO] [1718197599, 865980690]: hello world
[INFO] [1718197599, 885495469]: hello world
[INFO] [1718197599, 96461931]: 280
[INFO] [1718197599, 966797174]: hello world
[INFO] [1718197600, 000224081]: 281
[INFO] [1718197600, 668224161]: hello world
[INFO] [1718197600, 181922161]: 282
[INFO] [1718197600, 199391548]: hello world
[INFO] [1718197600, 267715863]: 283
[INFO] [1718197600, 370751134]: hello world
[INFO] [1718197600, 390038749]: hello world
[INFO] [1718197600, 47261529]: 285
[INFO] [1718197600, 474955467]: hello world
INF

[INFO] [1718197599, 486231433]: Recieved [275]
[INFO] [1718197599, 506853801]: Recieved greeting [hello world]
[INFO] [1718197599, 569738946]: Recieved greeting [hello world]
[INFO] [1718197599, 666290133]: Recieved greeting [hello world]
[INFO] [1718197599, 784533704]: Recieved greeting [hello world]
[INFO] [1718197599, 784669322]: Recieved greeting [hello world]
[INFO] [1718197599, 838803136]: Recieved greeting [hello world]
[INFO] [1718197599, 969446653]: Recieved greeting [hello world]
[INFO] [1718197599, 969579786]: Recieved [280]
[INFO] [1718197600, 071751661]: Recieved greeting [hello world]
[INFO] [1718197600, 22061518]: Recieved greeting [hello world]
[INFO] [1718197600, 222078572]: Recieved [282]
[INFO] [1718197600, 277098247]: Recieved greeting [hello world]
[INFO] [1718197600, 277426985]: Recieved [283]
[INFO] [1718197600, 39725471]: Recieved greeting [hello world]
[INFO] [1718197600, 519687985]: Recieved greeting [hello world]
[INFO] [1718197600, 53725508]: Recieved [285]
[INFO] [1718197600, 568699814]: Recieved greeting [hello world]
[INFO] [1718197600, 568991276]: Recieved [286]

Berikut merupakan setelah mengikuti bagian msg publisher dan subscriber

Ubuntu 20.04 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Jun 12 20:24

CMakeLists.txt

```
bob@root:~/bobe_ws
```

111 ## CATE
112 ## catkin
113 catkin:
114 # INC[10%] Building CXX object mastering_ros_demo_pkg/CMakeFiles/demo_service_client
115 [178198671.698532371]: From Client [Sending from Here], Server says [Re
ceived Here]
116 # CAT[10%] Built target _mastering_ros_demo_pkg_generate_messages_check_deps_demo_sr
117 ## DEPVN
118 CATE[25%] Generating Javascript code from mastering_ros_demo_pkg/demo_srv.srv
119 [178198671.797536856]: From Client [Sending from Here], Server says [Re
ceived Here]
120 [178198671.92383403]: From Client [Sending from Here], Server says [Re
ceived Here]
121 #####[41%] Generating Euslisp code from mastering_ros_demo_pkg/demo_srv.srv
122 ## Built[50%] Built target mastering_ros_demo_pkg_generate_messages_eus
123 #####[58%] Generating C code from mastering_ros_demo_pkg/demo_srv.srv
124 [178198671.934782442]: From Client [Sending from Here], Server says [Re
ceived Here]
125 ## Spec[66%] Generating Python code from Srv mastering_ros_demo_pkg/demo_srv.srv
126 ## Your[75%] Generating Python srv __init__.py for mastering_ros_demo_pkg
127 [178198672.018525970]: From Client [Sending from Here], Server says [Re
ceived Here]
128 # Include[83%] Generating Python msg __init__.py for mastering_ros_demo_pkg
129 # Cate[91%] Generating message files for mastering_ros_demo_pkg_generate_messages_py
130 [178198672.51416315]: From Client [Sending from Here], Server says [Re
ceived Here]
131 ## Linking CXX executable /home/bobe/bobe_ws/devel/lib/mastering_ros_demo_pk
132 ## add[91%] Built target demo_service_client
133 ## add[100%] Linking CXX executable /home/bobe/bobe_ws/devel/lib/mastering_ros_demo_pk
134 ## add[100%] Built target demo_service_server
135 ## add[100%] Built target demo_service_server
136 ## add[100%] Built target demo_service_server
137 ## add[100%] bobe@root:~/bobe_ws \$
138 ## add_executable(demo_service_client src/demo_service_client.cpp)
139 ## add_executable(demo_service_server src/demo_service_server.cpp)
140
141 #target link_libraries(demo_topic_publisher \${catkin_LIBRARIES})
142 #target link_libraries(demo_topic_subscriber \${catkin_LIBRARIES})
143 #target link_libraries(demo_msg_publisher \${catkin_LIBRARIES})
144 #target link_libraries(demo_msg_subscriber \${catkin_LIBRARIES})
145 #target link_libraries(demo_action_client \${catkin_LIBRARIES})
146 #target link_libraries(demo_action_server \${catkin_LIBRARIES})
147 target_link_libraries(demo_service_client \${catkin_LIBRARIES})
148 target_link_libraries(demo_service_server \${catkin_LIBRARIES})
149
150 ## Declare a c++ library
151 ## add_library(\${PROJECT_NAME})
152 ## src \${PROJECT_NAME}/mastering_ros_demo_pkg.cpp
153 ## #
154
155 ## Add cmake target dependencies of the library
156 ## as an example, code may need to be generated before libraries
157 ## either from message generation or dynamic reconfigure
158 ## add_dependencies(\${PROJECT_NAME} \${CMAKE_PROJECT_NAME}_GENERATED_TARGETS)

Save

bobe@root:~

[INFO] [178198671.698532371]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198671.797536856]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198671.92383403]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.018525970]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.140857878]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.212417089]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.343782442]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.418256052]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.51416315]: From Client [Sending from Here], Server says [Re
ceived Here]
[INFO] [178198672.637107649]: From Client [Sending from Here], Server says [Re
ceived Here]
[INF

Berikut merupakan setelah mengikuti bagian service server dan client

The screenshot shows a Linux desktop environment with several windows open. The terminal window displays the build logs for a CMake project, specifically for the 'mastering_ros_demo_pkgs' package. The logs show the compilation of various files (C++, Python, Lisp) and the linking of executables. A message indicates that the 'demo_action_client' node has failed to call service. Another terminal window shows the roscore logs, which include messages from the 'rospy' library and the 'roslaunch' command starting the 'Demo Action Server'. The desktop background is a light blue gradient, and the taskbar at the bottom shows icons for various applications like a browser, file manager, and system monitor.

```
Ubuntu 20.04 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 13 18:08
Jun 13 18:08
package.xml
+/bobe_ws/cmake/mastering_ros_demo_pkg
package.xml
[ INFO] [1718198705.647411080]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.750167188]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.835123951]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.977917970]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.040348546]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.158298044]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.282997269]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.428989574]: Starting Demo Action Server
[ INFO] [1718276983.923509034]: /demo_action is processing the goal 10
[ INFO] [1718276983.940455957]: Setting to goal 0 / 10
[ INFO] [1718276984.138631751]: Setting to goal 1 / 10
[ INFO] [1718276984.333682751]: Setting to goal 2 / 10
[ INFO] [1718276984.533683403]: Setting to goal 3 / 10
[ INFO] [1718276984.733253383]: Setting to goal 4 / 10
[ARN] [1718276904.929899574]: /demo_action got preempted!
[ INFO] [1718198705.542196087]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.652196087]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.650497830]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.757492815]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.836854385]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198705.983436080]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.043436076]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.172901169]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.304519846]: From Client [Sending from Here], Server says [Received Here]
[ INFO] [1718198706.457909389]: Failed to call service
[bobe@root:~]$ ros runmaster mastering_ros_demo_pkg demo_action_client 10
[ INFO] [1718276983.928147126]: Action server started, sending goal.
[ INFO] [1718276983.928212925]: Sending Goal [10] and Preempt time of [1]
[ INFO] [1718276984.929370636]: Action did not finish before the time out.
[INFO] [1718276984.929370636]: halting now - 5
```

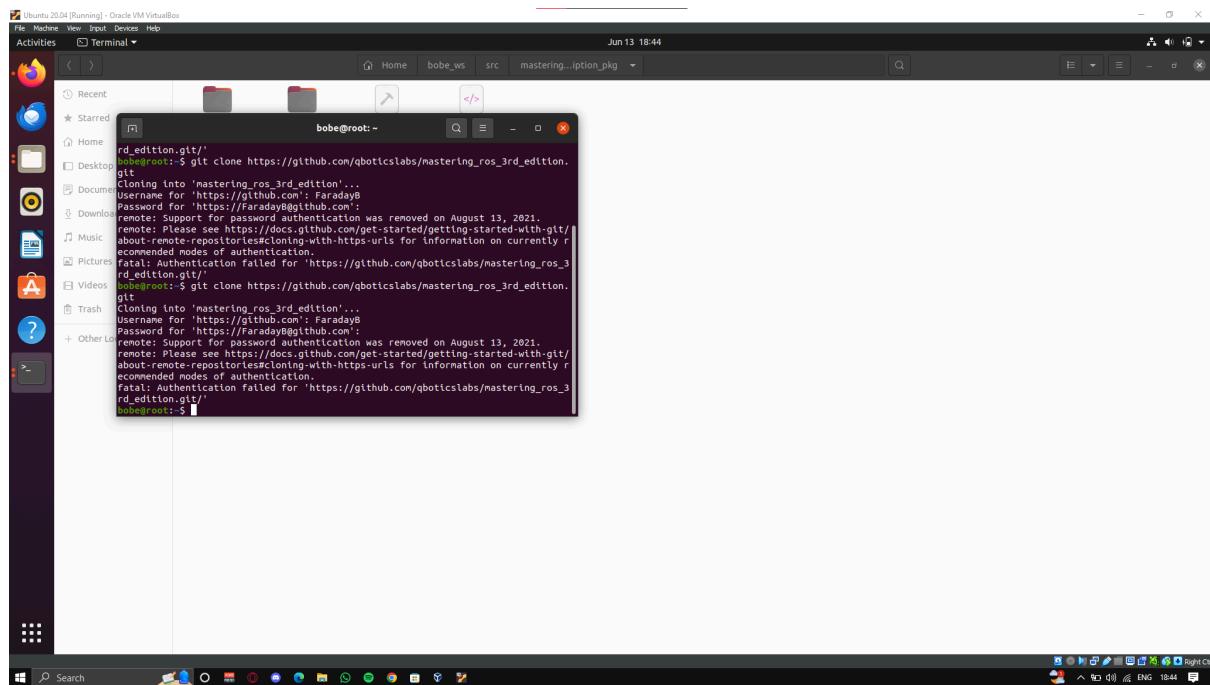
Berikut merupakan setelah mengikuti bagian action server dan client

Gambar-gambar diatas merupakan salah contoh-contoh untuk membuat dan menjalankan program ROS. Yang pertama adalah kita membuat sebuah ROS Packages atau basic unit dari sebuah program ROS. Paket ROS ini yang akan digunakan oleh robot untuk melakukan aksi yang telah ditentukan di paket tersebut. Kemudian, disini juga terlihat beberapa contoh untuk membuat topics sebagai alat komunikasi antar ROS Nodes, membuat custom messages untuk komunikasi, service yang digunakan untuk robot tersebut, dan membuat action untuk mendapatkan goal, feedback, dan result dari robot.

BAB 3

Pada bab ini, akan dipelajari cara untuk menggunakan ROS sebagai aplikasi 3D Modelling. Selain itu juga akan mengetahui apa itu Unified Robot Description Format (URDF) sebagai file model robot yang telah dibuat. Namun, saya tidak dapat melakukan percobaan tersebut karena github yang tidak dapat diakses karena beberapa fitur github yang sudah dihapus. Walaupun begitu, kita dapat melihat di buku tersebut cara untuk membuat robot tersebut dengan kode ketentuan yang sudah diberikan. Intinya, URDF adalah tipe file yang digunakan apabila kita telah membuat model robot di RVIZ (Robot Visualization). Dari hasil yang telah kita buat, kita dapat memvisualisasikannya pada aplikasi ROS tersebut, yaitu RVIZ. Selain itu juga cara kerja dari link yang biasanya digunakan sebagai sendi dari bagian robot tersebut.

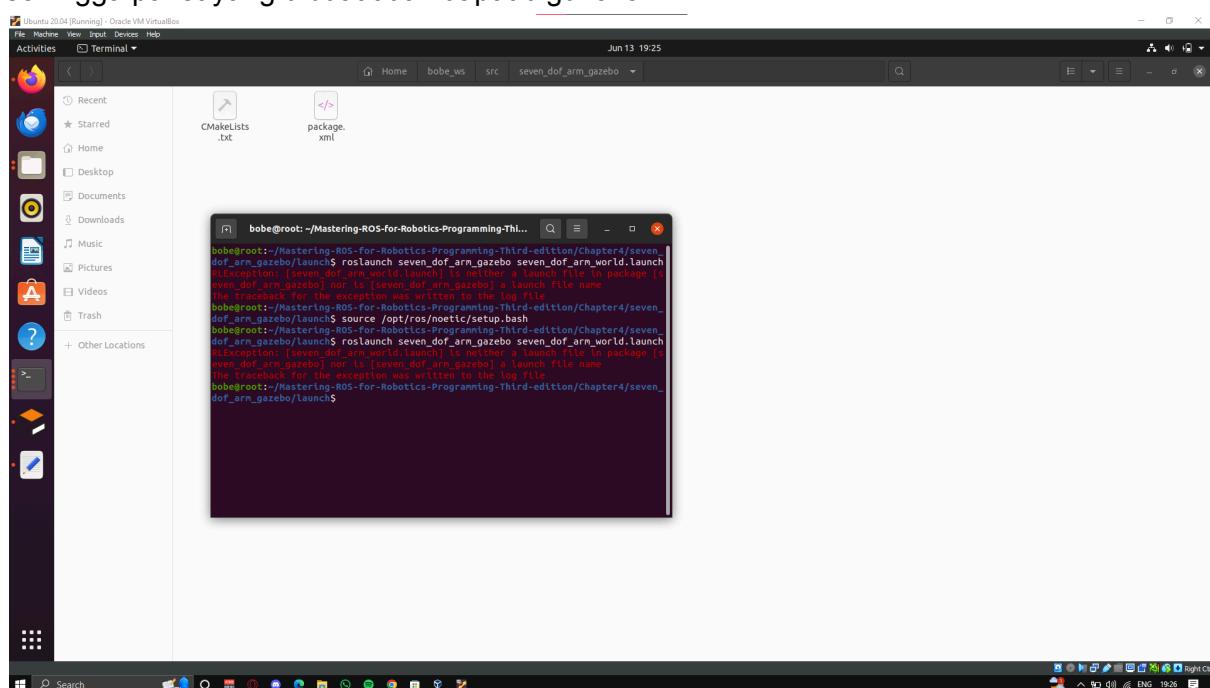
Bentuk URDF biasanya sangat umum dipelajari di awal saat ingin membuat model robot. Namun, beberapa fitur yang tidak adalah kemudahan, penggunaan kembali, modularitas, dan mudahnya di program. Maka dari itu, ada bentuk file lain dari model robot, yaitu xacro. Xacro sendiri sudah menjadi solusi dari beberapa fitur yang tidak ada di URDF, yaitu menyederhanakan URDF dan mudahnya untuk di program. Bentuk xacro dapat diubah menjadi bentuk URDF apabila diperlukan.



```
Ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 13 18:44
Home bobo_ws src mastering...ption_pkg
bobe@root:~$ git clone https://github.com/qboticslabs/mastering_ros_3rd_edition.git
Cloning into 'mastering_ros_3rd_edition'...
Username for 'https://github.com': Faraday
Password for 'https://Faraday@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/qboticslabs/mastering_ros_3rd_edition.git'
bobe@root:~$
```

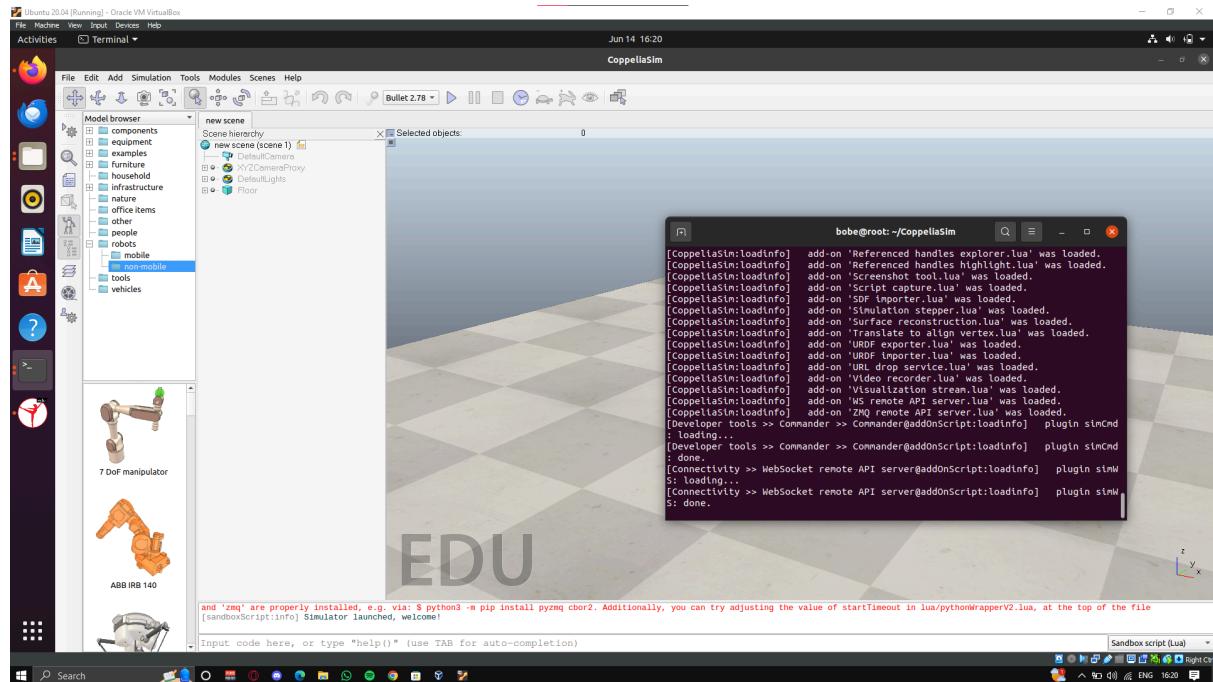
BAB 4

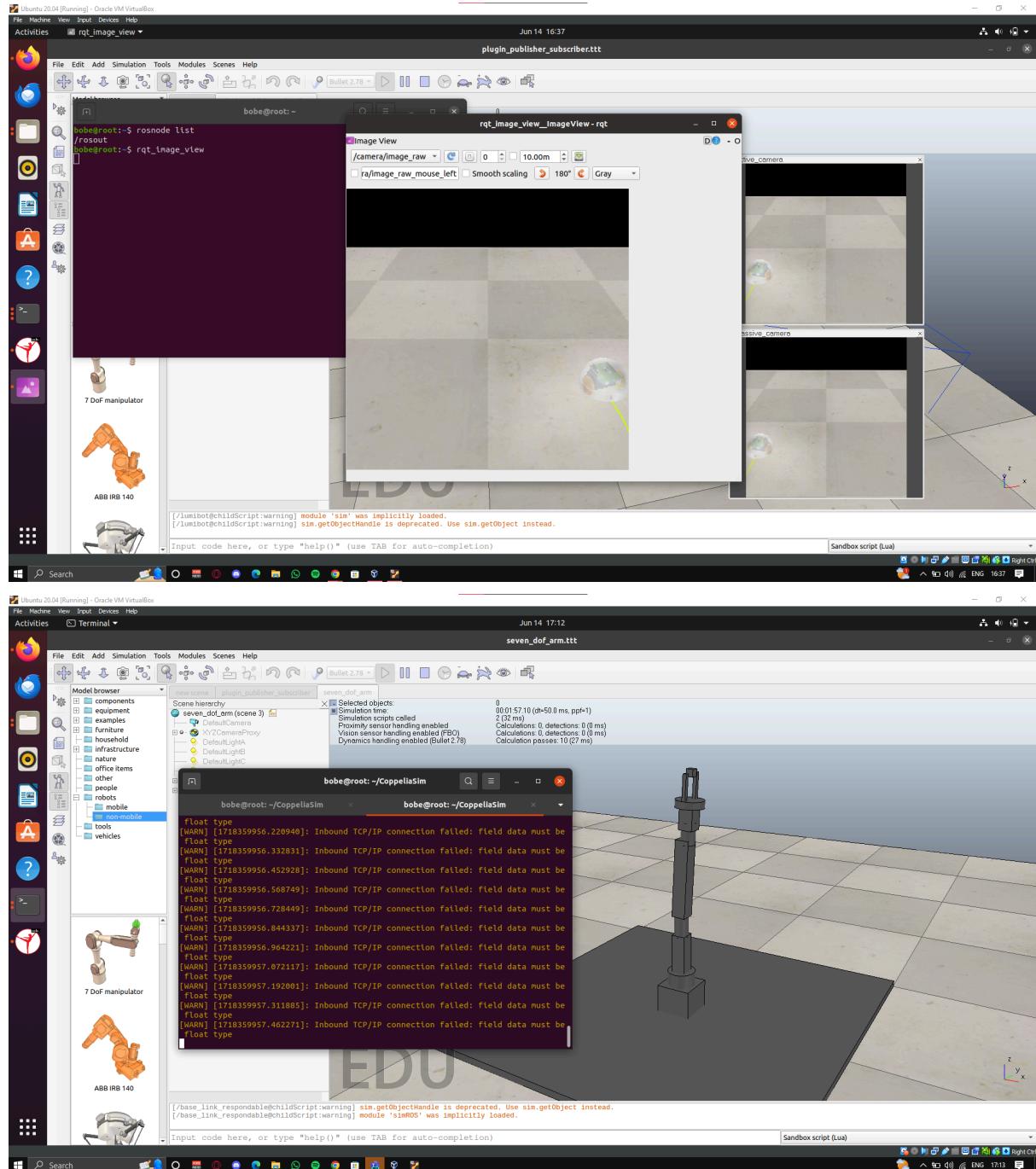
Pada bab ini, akan mempelajari simulator robot bernama Gazebo. Aplikasi ini adalah aplikasi bawaan apabila mengunduh ROS dan digunakan mensimulasikan robot yang telah di buat pada bagian sebelumnya. Disini akan ada beberapa contoh seperti menjalankan simulasi robot, menggerakan robot tersebut, dan ada contoh lain yang diberikan. Sayangnya saya tidak dapat menjalankan bagian ini karena merupakan lanjutan dari bab sebelumnya sehingga paket yang dibuat tidak dapat digunakan.



```
Ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 13 19:25
Home bobo_ws src seven_dof_arm_gazebo
bobe@root:~/Mastering-ROS-for-Robotics-Programming-Third-edition/Chapter4/seven_dof_arm_gazebo$ roslaunch seven_dof_arm_gazebo seven_dof_arm_world.launch
[roslaunch] ERROR: package[sseven_dof_arm_gazebo] not found
[roslaunch] ERROR: package[sseven_dof_arm_gazebo] not ls[sseven_dof_arm_gazebo] a launch file name
The traceback for the exception was written to the log file
bobe@root:~/Mastering-ROS-for-Robotics-Programming-Third-edition/Chapter4/seven_dof_arm_gazebo$ source /opt/ros/noetic/setup.bash
bobe@root:~/Mastering-ROS-for-Robotics-Programming-Third-edition/Chapter4/seven_dof_arm_gazebo$ roslaunch seven_dof_arm_gazebo seven_dof_arm_world.launch
[roslaunch] ERROR: package[sseven_dof_arm_world.launch] ls neither a launch file in package [seven_dof_arm_gazebo] nor ls[seven_dof_arm_gazebo] a launch file name
The traceback for the exception was written to the log file
bobe@root:~/Mastering-ROS-for-Robotics-Programming-Third-edition/Chapter4/seven_dof_arm_gazebo$
```

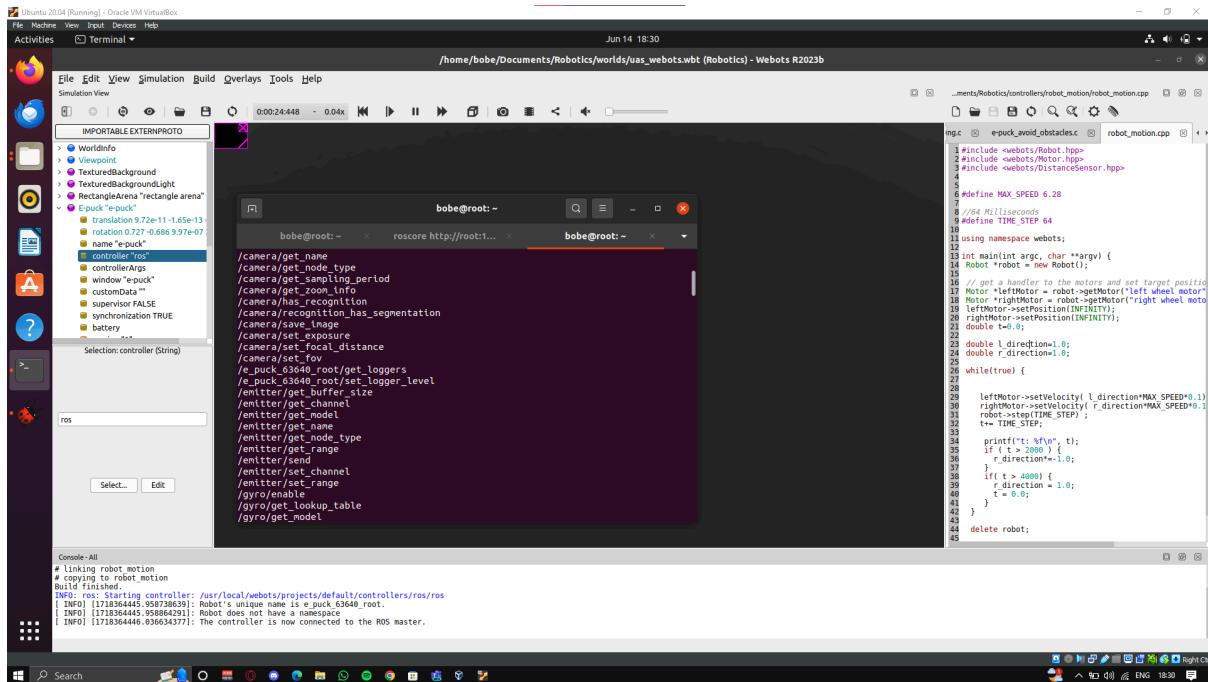
BAB 5



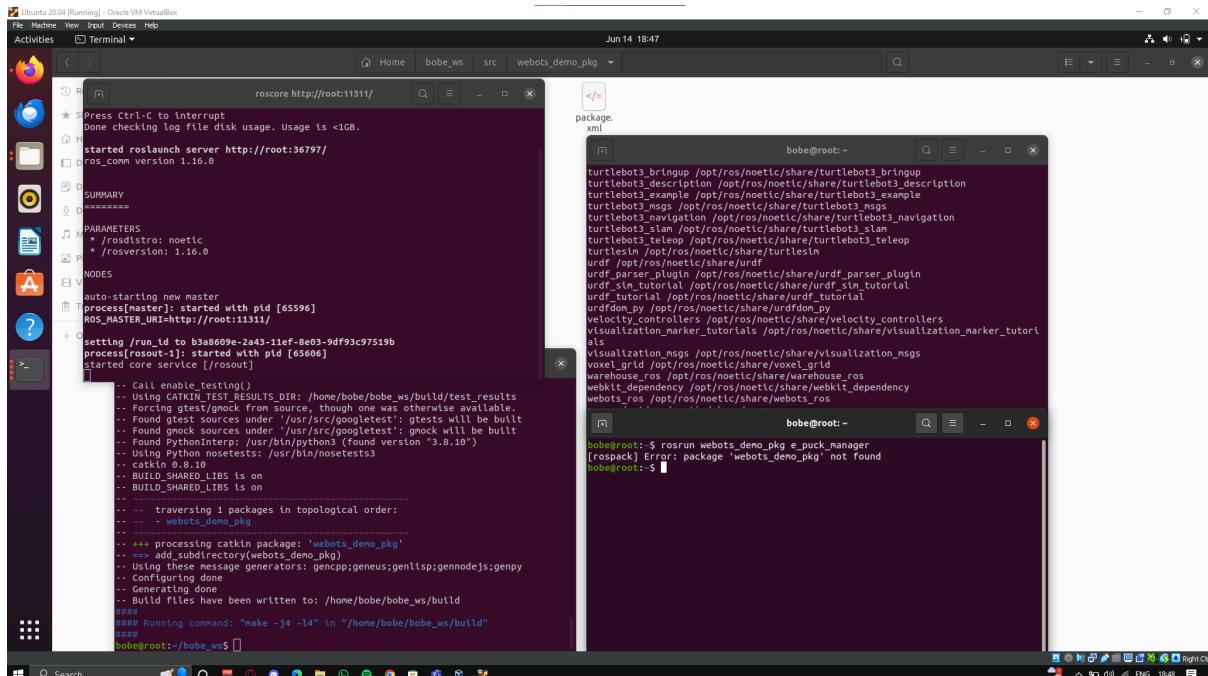


Berikut merupakan hasil setelah install Coppelia Sim versi edu

Pada bagian ini, kita belajar cara menggunakan Coppelia Sim sebagai simulator robot untuk menjalankan file .ttt untuk menjalankan file yang sama seperti xacro dan urdf. Selain itu, Coppelia Sim bisa menggunakan Remote API yang berfungsi sebagai komunikasi yang menggunakan client dan server. Client digunakan sebagai aplikasi eksternal untuk implementasi node ROS. Sedangkan server digunakan untuk mendapatkan script sehingga dapat mendapatkan data untuk berinteraksi dengan simulasi tersebut. Pada 2 gambar pertama di atas, terdapat file yang dapat dijalankan untuk simulator tersebut, namun pada gambar ke-3 terdapat command yang tidak dapat dijalankan sehingga tidak dapat melanjutkan tahap-tahap berikutnya.



Selanjutnya kita menggunakan webots untuk menggunakan ROS. Di buku ini menjelaskan cara untuk menginstall Webots pada ubuntu dan menggunakannya dengan ROS. Kita juga diajarkan basic untuk menggunakan webots seperti menambahkan robot, membuat kode untuk robot, dan mengubah kode yang digunakan oleh robot. Namun dalam mengaplikasikannya, saya tidak dapat menjalankannya yang ada di video karena ternyata tidak ada opsi untuk menjalankannya. Gambar diatas merupakan hasil yang didapatkan.



Kemudian dilakukan lagi penggunaan webots untuk menjalankan file CPP untuk robot e-puck, disini seharusnya menjalankan sebuah file controller agar dapat menjalankan melihat e-puck dan mengontrol kameranya. Namun masalah yang saya alami adalah package tersebut tidak dapat dibuat sehingga saya tidak dapat menjalankannya.

BAB 6

Pada bab ini, ada beberapa yang kita pelajari, yaitu node move_group, motion_planning menggunakan MoveIt!

Node move_group adalah komponen utama dari MoveIt!, mengintegrasikan berbagai bagian robot dan menyediakan tindakan/layanan berdasarkan kebutuhan pengguna. Node ini mengumpulkan informasi robot, seperti point cloud, status gabungan, dan transformasi, dari topik dan layanan. Ini juga memperoleh data kinematika robot, URDF, SRDF, dan file konfigurasi, dari server parameter.

Setelah memiliki semua informasi yang diperlukan, pengguna dapat memerintahkan node move_group untuk melakukan tindakan menggunakan C++ atau Python MoveIt! API. Plugin perencanaan gerak RViz memungkinkan memerintahkan robot dari GUI RViz.

Node move_group bertindak sebagai integrator dan menghubungkan fungsionalitas melalui plugin untuk pemecah kinematika dan perencanaan gerak. Setelah perencanaan gerakan, lintasan yang dihasilkan dikirim ke pengontrol menggunakan antarmuka FollowJointTrajectoryAction untuk dieksekusi pada robot atau simulator nyata.

MoveIt! dapat dihubungkan ke RViz GUI ke Gazebo untuk mengendalikan lengan robot. Perencanaan gerak adalah teknik yang digunakan untuk menemukan jalur optimal bagi robot untuk bergerak dari pose awal ke pose tujuan tanpa menabrak rintangan apa pun. Hal ini membutuhkan deskripsi robot dan geometri dunia, yang dapat direpresentasikan menggunakan file URDF.

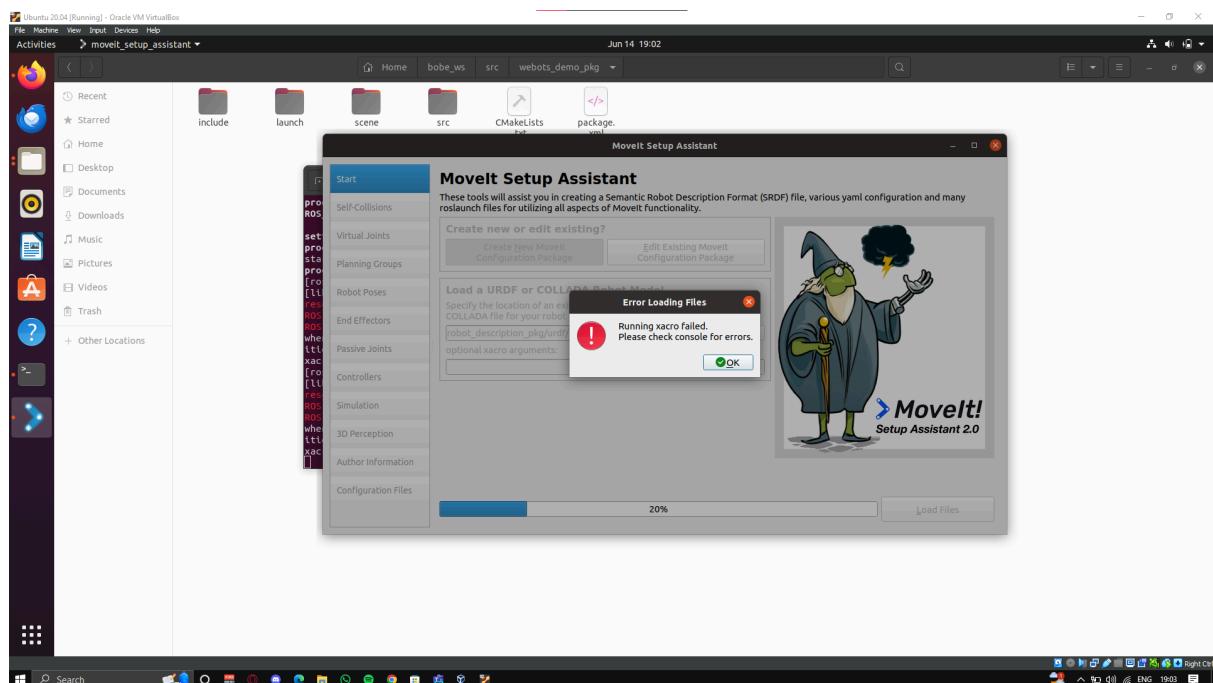
Kendala dapat ditambahkan ke perencana gerakan, seperti posisi, orientasi, visibilitas, batas sendi, atau kendala yang ditentukan pengguna. Adaptor permintaan perencanaan gerak melakukan praproses dan pascaproses permintaan dan respons perencanaan gerak.

Adegan perencanaan di MoveIt! merepresentasikan dunia di sekitar robot dan menyimpan status robot. Ia membaca keadaan bersama dan informasi sensor untuk membangun geometri dunia, termasuk octomap yang dihasilkan dari point cloud atau gambar kedalaman.

MoveIt! memberikan fleksibilitas untuk mengganti algoritma IK menggunakan plugin dan sudah mengintegrasikan FK dan menemukan Jacobian. Pemeriksaan tabrakan di MoveIt! dilakukan dengan menggunakan objek CollisionWorld dengan dukungan untuk berbagai jenis objek. Matriks Tabrakan yang Diizinkan (ACM) digunakan untuk mengoptimalkan komputasi pemeriksaan tabrakan.

Untuk menghubungkan lengan robot dengan MoveIt!, simpul move_group memerlukan URDF, SRDF, file konfigurasi, dan topik status gabungan bersama dengan informasi TF. Alat Bantu Pengaturan di MoveIt! dapat digunakan untuk menghasilkan semua elemen ini untuk paket konfigurasi MoveIt!

Untuk menjalankan MoveIt Assistant ini terdapat kendala yang dialami, yaitu seperti gambar dibawah ini.



Masalah yang dialami adalah tidak dapat mendeteksi file xacro yang ada walaupun memang sudah terunduh.