

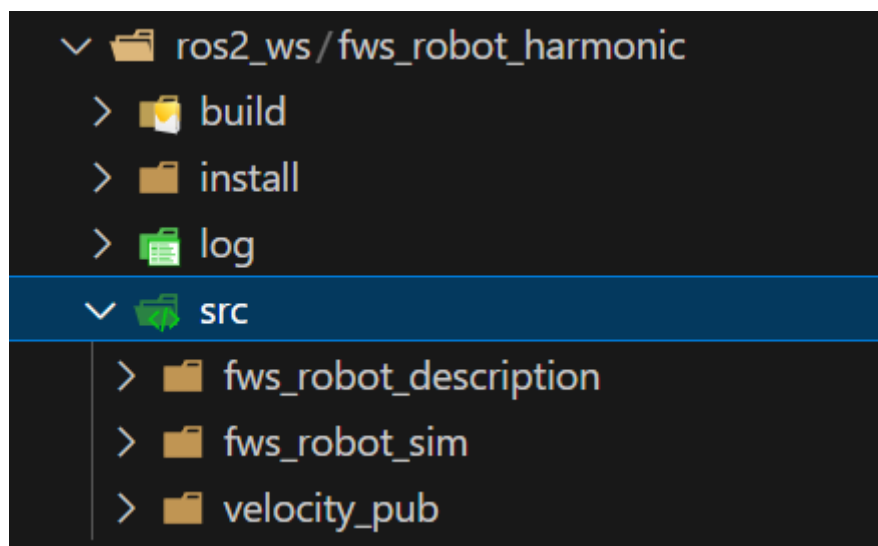
Laporan Tubes Programming URO

Aryo Wisanggeni

13523100

Menggunakan Gazebo dalam ROS2

Sudah dua hari mengotak-atik Gazebo dan ROS2 dan simulasinya tidak dapat dilakukan. Hal ini karena versi Ubuntu yang digunakan berupa 24.04 atau noble yang hanya support ROS2 rolling dan jazzy. Setelah instalasi jazzy dan mencoba mendownload gazebo, gazebo yang berhasil didownload adalah gazebo sim, bukan gazebo classic. Video dan resource untuk gazebo sim menurut saya kurang membantu dan sulit untuk dijalankan. Berikut file structure dari project ROS2:



Gambar 1. File structure dari project ROS2

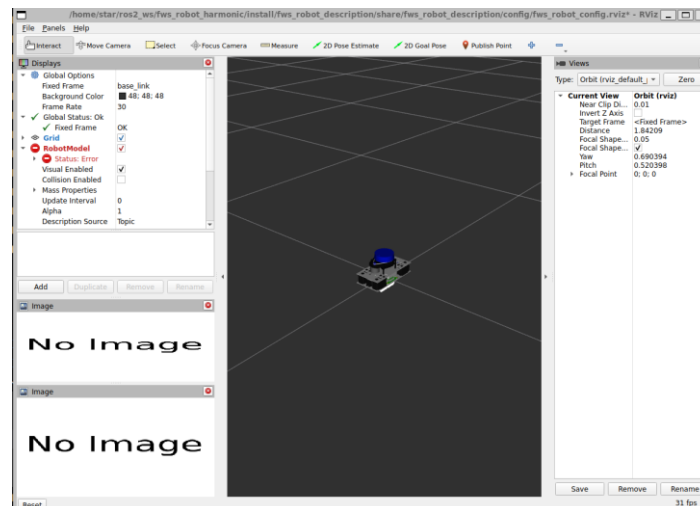
Di sini folder terpenting adalah folder src. Folder ini terdiri dari tiga folder lagi yang mempunyai fungsinya masing-masing.

1. Fws_robot_description berfungsi untuk menyimpan informasi mengenai robot yang digunakan, seperti file .urdf.xacro dari robot utama dan komponen-komponennya, serta file meshes dari komponen-komponen tersebut
2. Fws_robot_sim berfungsi untuk menyimpan script untuk launch simulasi dan juga file world dari simulasi yang berekstensi .world atau .sdf
3. Terakhir adalah velocity_pub yang menyimpan script untuk mengontrol robot.

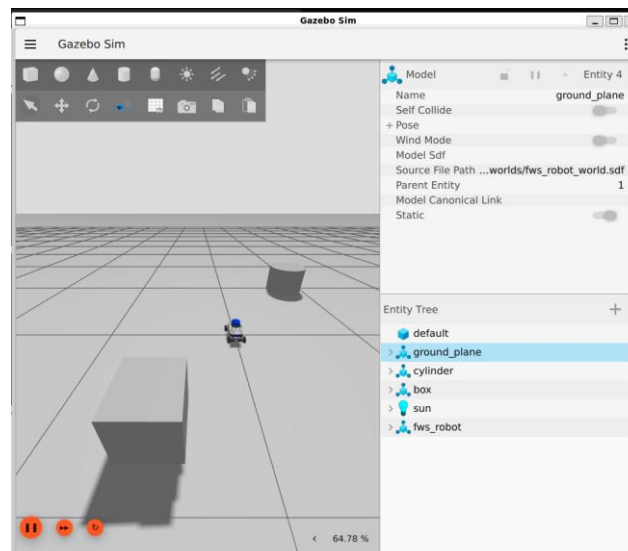
Folder lainnya (build, install, dan log) dihasilkan dari command *colcon build*. Selain itu untuk menjalankan simulasi harus dijalankan command-command berikut sesuai urutan.

```
colcon build
source /opt/ros/jazzy/setup.bash
source fws_robot_harmonic/install/setup.bash
ros2 launch fws_robot_sim fws_robot_spawn.launch.py
```

Berikut merupakan gambar menjalankan command-command di atas.



Gambar 2. Simulasi Rviz yang dijalankan



Gambar 3. Simulasi Gazebo yang dijalankan

Walaupun sempat dijalankan gazebo nya, saya tidak berhasil memobilisasi robotnya.

Menggunakan Unity Untuk Mensimulasi Algoritma A*

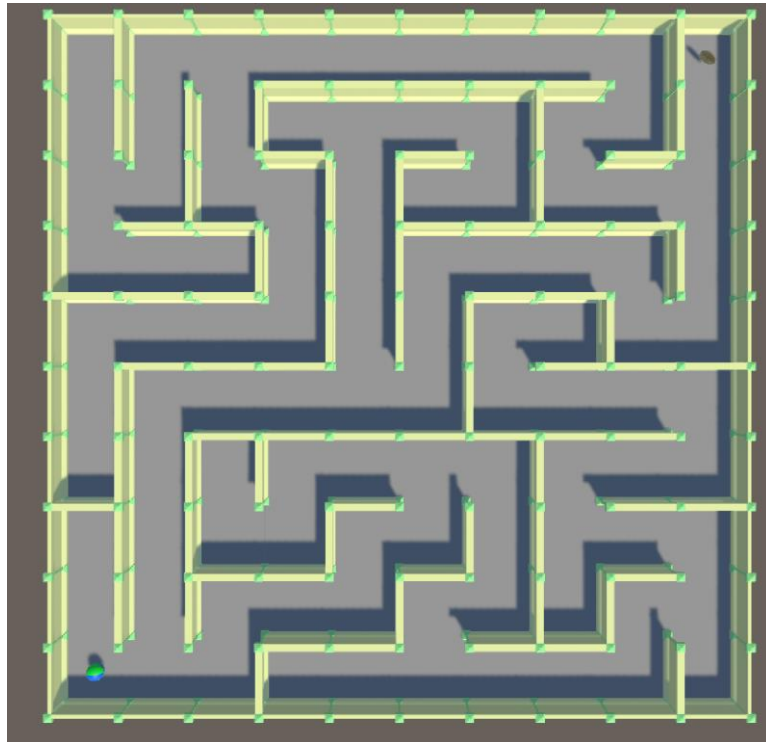
A* adalah algoritma untuk mencari jalur terpendek dari satu node ke node lain dari suatu graf. Karena robot tidak berhasil dijalankan di gazebo, Algoritma A* akan disimulasi menggunakan unity.

Program yang dibuat untuk mendemonstrasikan algoritma ini, menggunakan mekanisme sebagai berikut:

1. Generate labyrinth baru yang randomized dengan algoritma rekursi tree
2. Cetak labyrinth tersebut ke suatu matrix yang akan digunakan untuk mencari jalur terpendeknya.
3. Gunakan algoritma A* pada matrix tersebut untuk mencari jalur terpendek

- Gunakan jalur terpendek itu untuk menggerakkan objek ke goal.

Pada repository terdapat video simulasi program ini yang dapat ditonton.



Gambar 4. Contoh labyrinth yang degenerate dengan cell 10x10

Untuk implementasi programnya, labyrinth dengan cell $N \times N$ akan menjadi matrix $(2N+1) \times (2N+1)$. Jadi, untuk labyrinth 3×3 , program akan menghasilkan matrix 7×7 seperti dibawah ini

#	X	#	X	#	X	#
X	O	-	O	X	G	X
#	X	#	-	#	-	#
X	O	X	O	-	O	X
#	-	#	X	#	-	#
X	B	-	O	-	O	X
#	X	#	X	#	X	#

Gambar 5. Contoh matrix dari labyrinth 3×3

Berikut makna dari berbagai karakter pada matrix ini:

- '#' adalah karakter filler, hanya untuk mengisi matrix
- 'B' adalah posisi mulai objek yang bergerak
- 'G' adalah gol/tujuan objek
- 'X' adalah tembok
- '-' adalah jalan yang bisa dilewati objek