

Nama : Muhammad Sidqi Nabhan
NIM : 1103200179
Kelas : TK 45 G09

Technical Report ROS2: Nodes Communication Using Services & Actions

A. ROS2

ROS 2, atau Robot Operating System 2, adalah platform perangkat lunak open source yang dirancang khusus untuk pengembangan dan pengoperasian robot. ROS 2 merupakan evolusi dari ROS yang sebelumnya (ROS 1), dan dirancang untuk mengatasi beberapa keterbatasan yang ada pada versi sebelumnya.

ROS 2 dirancang dengan dukungan untuk komunikasi real-time, yang merupakan perbaikan signifikan dari ROS 1. Ini memungkinkan pengembang untuk mengimplementasikan sistem yang lebih responsif dan dapat diandalkan.

ROS 2 dirancang dengan mempertimbangkan skalabilitas, yang memungkinkan untuk digunakan dalam berbagai konteks, termasuk robot-robot yang membutuhkan tingkat kompleksitas dan keandalan yang berbeda.

ROS 2 diterapkan dengan desain modular yang memungkinkan pengembang untuk menggunakan komponen-komponen yang diperlukan dan meninggalkan yang tidak diperlukan. Hal ini meningkatkan reusabilitas dan mempermudah pengembangan aplikasi.

ROS 2 mendukung beberapa bahasa pemrograman, termasuk C++, Python, dan lainnya. Ini memberikan fleksibilitas lebih lanjut kepada pengembang untuk memilih bahasa yang paling sesuai dengan keahlian mereka.

Meskipun merupakan versi yang berbeda, ROS 2 dirancang dengan mempertimbangkan kompatibilitas dengan ROS 1. Ini berarti bahwa beberapa kode ROS 1 dapat diintegrasikan ke dalam proyek ROS 2.

B. Nodes Communication Using Services & Actions

Nodes itu sendiri merujuk pada unit dasar perangkat lunak yang menjalankan tugas tertentu dalam sistem robotika. Dalam konteks ROS, komunikasi antara node dapat difasilitasi menggunakan layanan (services) dan tindakan (actions).

a. Layanan (Services)

Layanan adalah pola komunikasi di mana satu node (klien) meminta tugas tertentu dari node lain (server). Ini adalah mekanisme komunikasi satu-ke-satu yang bersifat sinkron

b. Tindakan (Actions)

Tindakan adalah bentuk komunikasi yang lebih canggih, cocok untuk tugas yang memakan waktu lama atau memerlukan umpan balik selama eksekusi. Ini adalah mekanisme komunikasi satu-ke-satu atau satu-ke-banyak yang bersifat asinkron

C. Langkah Langkah pengerjaan

- a. Melakukan pengecekan pada *virtualbox* apakah ROS2 sudah terinstall dan dapat dijalankan

Gunakan perintah :

```
ros2
```

- b. Menjalankan command untuk memunculkan turtlesim yang digunakan untuk komunikasi node nantinya

Gunakan perintah:

```
Ros2 run turtlesim turtlesim_node
```

- c. Menggunakan command

```
ros2 run turtlesim turtle_teleop_key
```

Untuk menggerakan kura-kura sesuai arah yang kita mau serta membuat kura kura bergerak secara rotasi

- d. Selanjutnya masuk kebagian service , dimana dengan menggunakan command

```
ros2 service list
```

atau

```
ros2 service list-t
```

Kita dapat melihat banyak service yang terdapat pada node untuk berkomunikasi dengan turtlesim tadi

- e. Selanjutnya memilih salah satu dari service list yaitu /spawn dengan menggunakan command

```
ros2 service call/spawn turtlesim/srv/spawn "{x:2,y:2, theta: 3.14}";
```

Dari source code diatas akan memunculkan kura kura baru sesuai dengan letak yang kita isi pada [x], [y], dan theta

- f. Selanjutnya masuk kebagian action, dimana dengan menggunakan command

```
ros2 action list
```

atau

```
ros2 action list-t
```

Dapat melihat banyak action yang terdapat pada node untuk berkomunikasi dengan turtlesim.

- g. Selanjutnya kita akan mencoba pada turtle bot1 yang merupakan turtle bot utama dengan menggunakan command

```
ros2 action send_goal/turtle/rotate_absolute turtlesim/action "{theta:3.14};  
-feedback
```

- h. Jika sudah di running maka bisa dilihat bahwa turlebot1 tadi akan bergerak memutar secara otomatis hingga nantinya berhenti sendiri karena menggunakan - feedback maka akan terlihat juga history atau log pergerakan turlebot1 dari awal hingga berhasil

- i. Untuk bagian terakhir kita bisa melihat parameter dari tutlebot ini menggunakan command

```
ros2 param list
```

- j. Selanjutnya running command

```
ros2 param get /turtlesim background_g
```

Untuk melihat value dari background turtlebot. Dan nantinya akan menampilkan background value dengan angka tertentu untuk warnanya

- k. Lanjutkan dengan command

```
ros2 param set /turtlesim background_g 200
```

Fungsinya untuk mengubah warna background biru muda

- l. Terakhir , menggunakan command

```
ros2 param dump/turtlesim
```

Untuk menyimpan editan turtlesim kita yang berwarna biru muda, setelah running code dapat dilihat editan turtlesim berhasil disimpan dan kita dapat membuka turtlesim berwarna biru muda tersebut nantinya ketika running turtlesim.