

JURNAL ROS

8 MARET 2022

COURSE : ROS BASICS IN 5 DAYS

DECONSTRUCTION

ROS merupakan singkatan dari robot operating system, seperti namanya, ROS merupakan sistem operasi robot, ROS tidak berdiri sendiri, namun berdiri ditopang oleh sistem operasi lain yaitu **linux**, dan diakses menggunakan terminal linux.

Tujuan mempelajari ROS adalah untuk memahami dasar fundamental dari suatu robot yang efisien dan praktis, pada jurnal ini, saya akan menuliskan apa yang saya pelajari selama mengikuti kursus online *The construct*, apa yang saya tuliskan dibawah ini tidak dapat diartikan sebagai tulisan formal, melainkan hanyalah jurnal pembelajaran saya selama belajar ros pada website *The construct*, semoga jurnal ini bisa berguna untuk pembelajaran anggota divisi programming banyubaramatna kedepannya

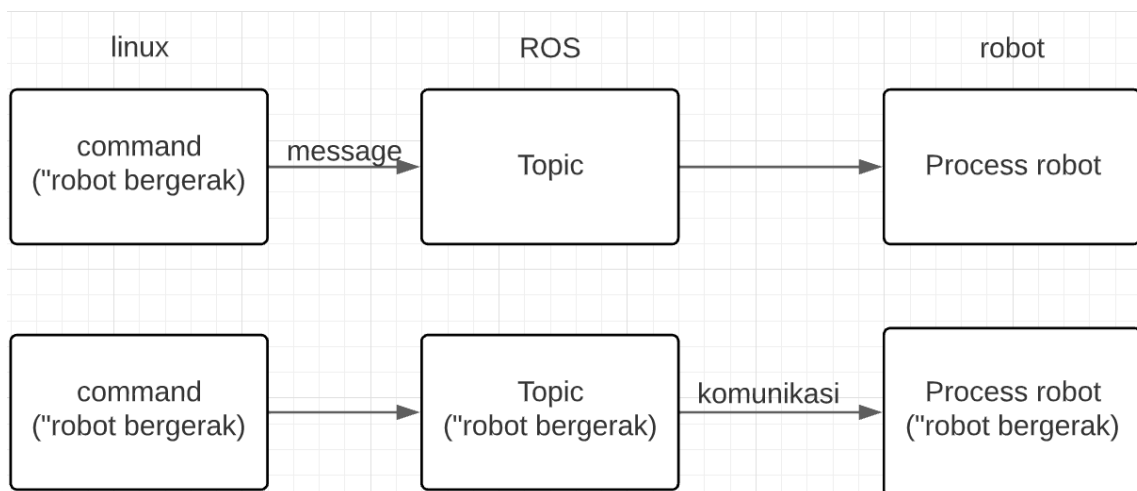
INSTALASI ROS

ROS sendiri merupakan sebuah package yang dapat diinstal pada sistem operasi linux, mengapa tidak windows ? alasannya sudah saya tuliskan pada jurnal sebelumnya, untuk penginstallan ros tidak akan saya cover pada jurnal ini, oleh karena itu pembaca dapat menginstall ros menggunakan link dibawah ini

<http://wiki.ros.org/ROS/Installation>

ROS TOPIC

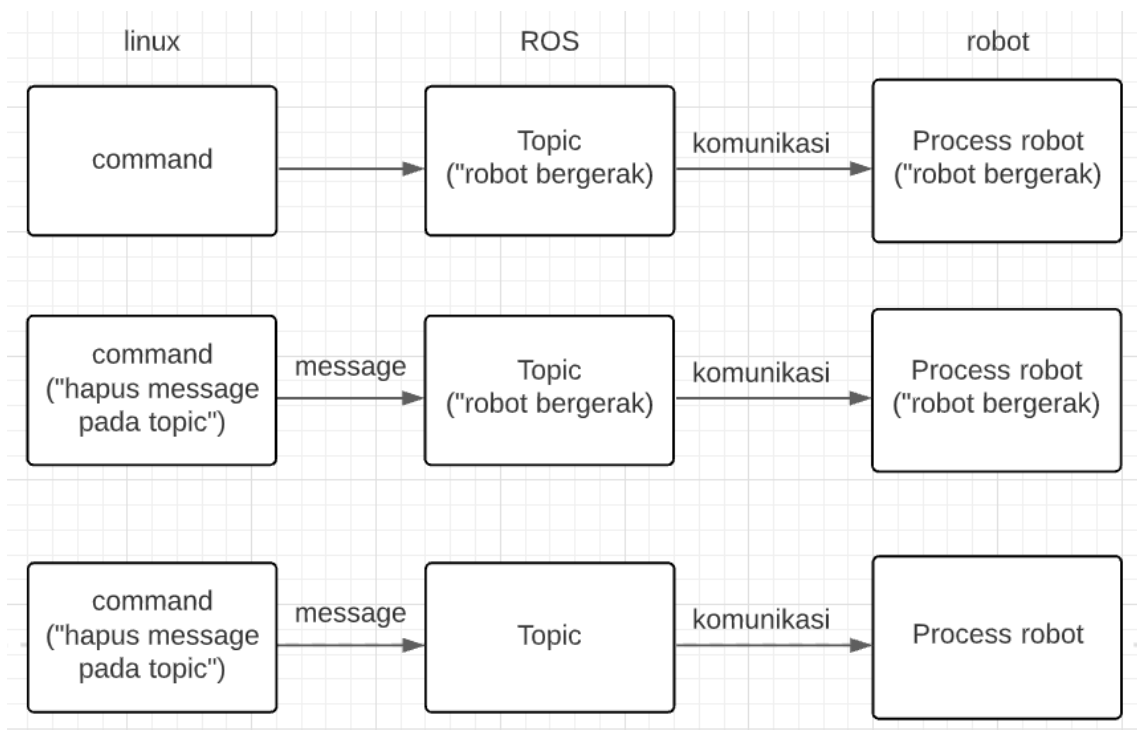
ROS dapat menjalankan perintah menggunakan **topic**, topic merupakan salah satu bagian terpenting dalam ROS, hal ini dikarenakan topic adalah bagian yang menangani seluruh komunikasi pada ROS, bahkan sistem komunikasi yang kompleks sekalipun, seperti **services** atau **action**, bergantung pada **topic**, dengan menggunakan topic, kamu dapat **berkomunikasi** dengan robotmu secara langsung, memberikan perintah secara langsung, seperti menyuruh robotmu bergerak, membaca hasil pembacaan sensor, dan lain lain.



Pada sesi Latihan *the construct*, diberikan command sebagai berikut:

```
roslaunch publisher_example move.launch
```

Command tersebut berfungsi untuk mengirimkan *messages* atau pesan kepada ros topic untuk menggerakkan robot (mohon diingat ini hanya contoh sederhana). Saat diberikan command ini, robot bergerak dan akan terus bergerak **selama command ini berjalan**, oleh karena itu untuk memamatikanya, diperlukan command pada keyboard yaitu **ctrl + c**.



Perlu diketahui juga, walaupun command untuk mengirimkan message telah dihentikan, message yang telah dikirimkan telah masuk pada topic , dan message tersebut akan terus dijalankan oleh robot selama message tersebut ada di dalam topic, nah maka dari itu , diperlukan sebuah **command yang dapat menghapuskan message** tersebut dari topic, yaitu sebagai berikut:

```
roslaunch publisher_example stop.launch
```

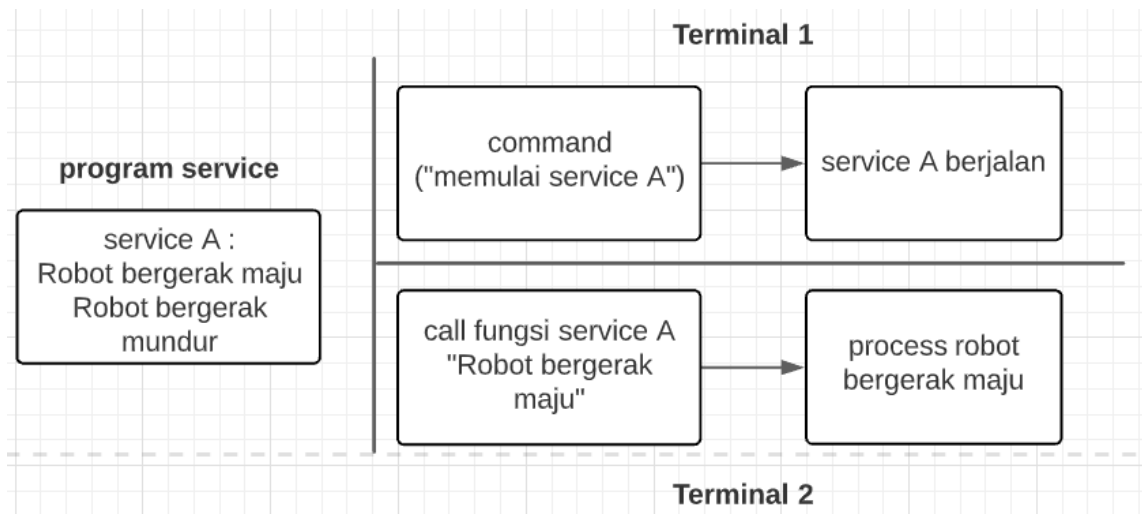
Maka dapat disimpulkan teori topic adalah sebagai berikut

- Topic, merupakan **perintah** pada ROS yang terus menerus **dibaca oleh robot**
- Untuk menjalankan topic, dapat menggunakan command
roslaunch

- Apabila command topic sudah diberikan, maka command tersebut akan terus **menyala hingga command tersebut dimatikan**
- Apabila command topic sudah dimatikan , **message** pada topic **akan terus menerus ada pada topic** hingga message tersebut **dihapus**

ROS SERVICE

Setelah membahas salah satu cara berkomunikasi pada robot menggunakan topic, saat ini saya akan membahas ros **service**, nah apa itu ros service ? ros service merupakan sebuah program built-in pada ros yang berfungsi untuk berkomunikasi pada robot juga, perbedaan ros service dengan ros topic , adalah berbeda dengan ros topic, ros service harus tetap aktif setiap waktu apabila ingin dipanggil fungsinya, berikut diagram alur sederhana untuk menggambarkan cara kerja ros service



Dapat dilihat, service memiliki dua sisi, yaitu **service center** dan **service client**. Pada diagram, service center diilustrasikan pada **terminal 1** dan service client diilustrasikan pada **terminal 2**.

Service center berfungsi sebagai tempat dimana **service berjalan** hingga dihentikan, sama seperti **topic** , service center dapat dihentikan dengan menekan tombol **ctrl + c** .

Service client adalah pelanggan dari service center , dimana **service client dapat memanggil salah satu atau lebih dari function dari service center**, seperti pada contoh , service client memanggil salah satu fungsi dari service center yaitu "robot bergerak maju" .

Maka, apa perbedaan topic dan service ? simple , **topic** akan terus menerus berjalan selama ada command di dalamnya hingga command tersebut di hapus, sementara **service** hanya akan berjalan sepanjang command berlangsung