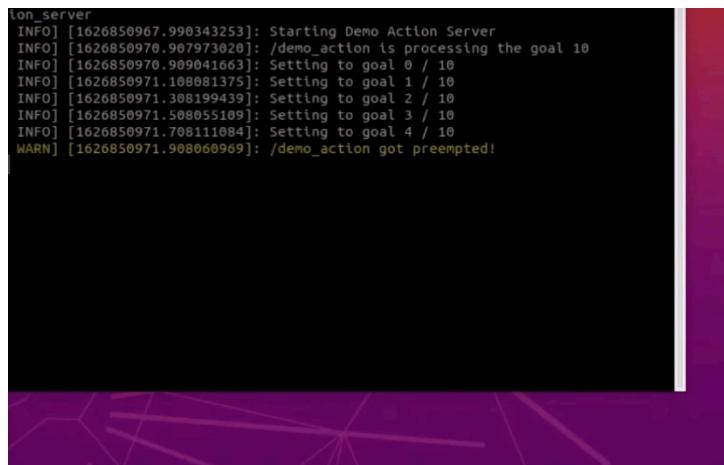


## 1. Bab 1: Introduction to ROS

Bab ini memberikan pengenalan menyeluruh tentang Robot Operating System (ROS), sebuah framework middleware yang digunakan untuk membangun aplikasi robotika yang kompleks. Penulis menjelaskan sejarah ROS, termasuk bagaimana dan mengapa sistem ini dikembangkan, serta peran penting yang dimainkan oleh ROS dalam menyederhanakan pengembangan perangkat lunak robotik dengan menyediakan berbagai alat dan pustaka yang diperlukan.

Selain itu, bab ini memandu proses instalasi ROS, memberikan instruksi langkah demi langkah yang mudah diikuti. Diperkenalkan konsep dasar ROS seperti node, topic, dan service, yang merupakan komponen inti dari ekosistem ROS. Node adalah unit proses yang menjalankan tugas tertentu, topic adalah saluran komunikasi antar-node, dan service adalah mekanisme untuk melakukan permintaan dan tanggapan antar-node.

## 2. Bab 2: Getting Started with ROS Programming



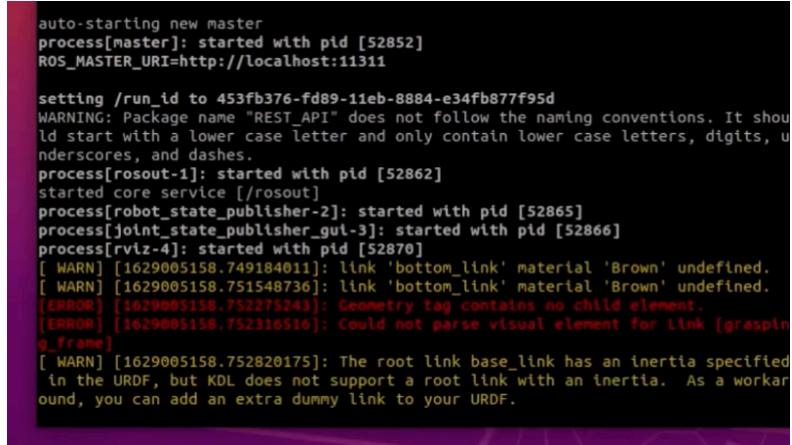
```
lon_server
[INFO] [1626850967.990343253]: Starting Demo Action Server
[INFO] [1626850970.907973020]: /demo_action is processing the goal 10
[INFO] [1626850970.909041663]: Setting to goal 0 / 10
[INFO] [1626850971.108081375]: Setting to goal 1 / 10
[INFO] [1626850971.308199439]: Setting to goal 2 / 10
[INFO] [1626850971.508055109]: Setting to goal 3 / 10
[INFO] [1626850971.708111084]: Setting to goal 4 / 10
[WARN] [1626850971.908066099]: /demo_action got preempted!
```

Bab ini menggali lebih dalam ke dalam arsitektur ROS dan konsep-konsep kuncinya. Dipelajari struktur jaringan ROS yang terdiri dari node, master, dan parameter server. Penulis menjelaskan bagaimana node berkomunikasi melalui model publish-subscribe dan client-server, serta bagaimana master bertindak sebagai pusat pendaftaran yang mengkoordinasikan komunikasi antar-node.

Di bagian ini, contoh praktis disediakan untuk membantu memahami implementasi konsep-konsep ini. Misalnya, belajar membuat node sederhana, mempublikasikan dan berlangganan ke topic, serta membuat dan memanggil service.

Dengan cara ini, bab ini tidak hanya memberikan teori tetapi juga aplikasi praktis yang dapat langsung dicoba.

### 3. Bab 3: Working with ROS Packages



```
auto-starting new master
process[master]: started with pid [52852]
ROS_MASTER_URI=http://localhost:11311

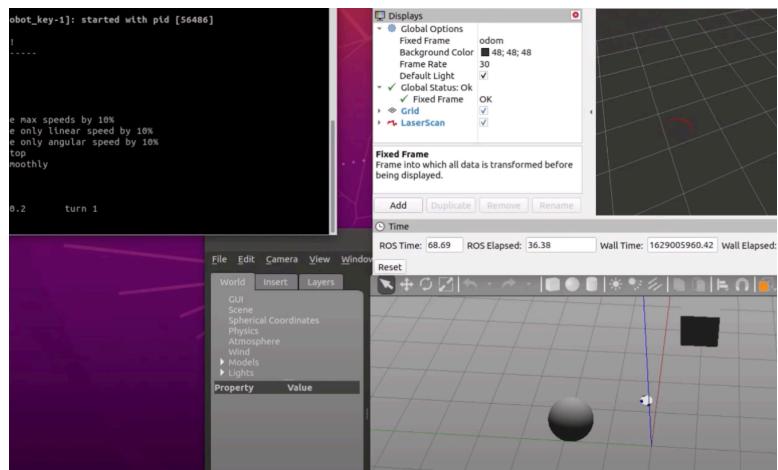
setting /run_id to 453fb376-fd89-11eb-8884-e34fb877f95d
WARNING: Package name "REST_API" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
process[rosout-1]: started with pid [52862]
started core service [/rosout]
process[robot_state_publisher-2]: started with pid [52865]
process[joint_state_publisher_gui-3]: started with pid [52866]
process[rviz-4]: started with pid [52870]
[ WARN] [1629005158.749184011]: link 'bottom_link' material 'Brown' undefined.
[ WARN] [1629005158.751548736]: link 'bottom_link' material 'Brown' undefined.
[ERROR] [1629005158.752275243]: Geometry tag contains no child element.
[ERROR] [1629005158.752316516]: Could not parse visual element for Link [graspin
g_frame]
[ WARN] [1629005158.752820175]: The root link base_link has an inertia specified
in the URDF, but KDL does not support a root link with an inertia. As a workar
ound, you can add an extra dummy link to your URDF.
```

Bab ini fokus pada paket-paket ROS yang merupakan blok bangunan utama dalam pengembangan aplikasi ROS. Penulis menjelaskan cara membuat, mengelola, dan menggunakan berbagai paket yang terdiri dari kode, data, dan dokumentasi yang dibutuhkan untuk proyek robotika. Dipelajari cara membuat paket baru, menambahkan dependensi, dan menggunakan sistem build catkin untuk membangun dan menguji paket.

Selain itu, bab ini mencakup penggunaan tools ROS seperti `roslaunch` untuk menjalankan aplikasi, dan `rosbag` untuk merekam dan memutar ulang data dari sensor. Penulis juga menunjukkan bagaimana mengintegrasikan berbagai paket yang sudah ada dengan paket yang dibuat, sehingga memudahkan pengembangan aplikasi yang lebih kompleks.

### 4. Bab 4: Simulating Robots Using ROS and Gazebo

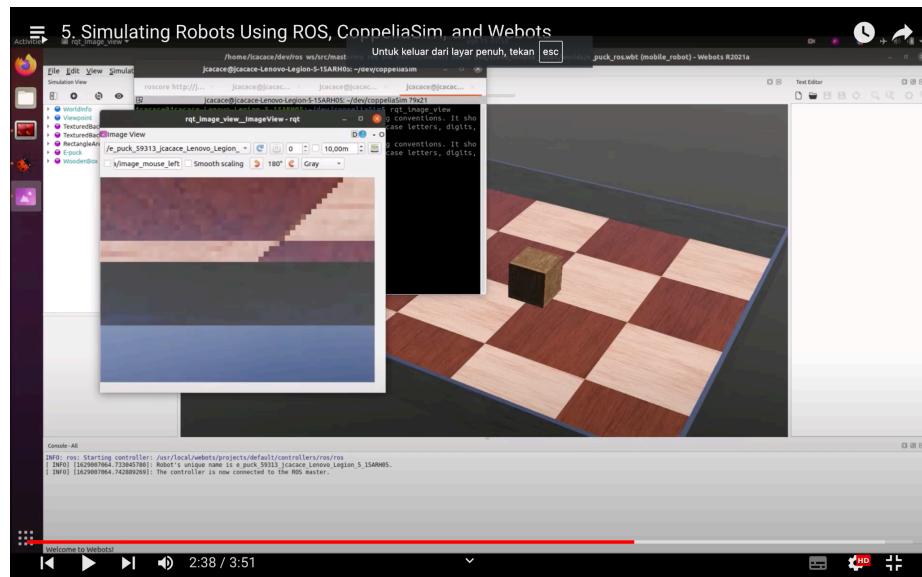
**TUGAS UAS ROBOTIKA**  
Alfian Mohamad Firdaus | 1103204002



Bab ini membahas penggunaan Gazebo, simulator robotik yang kuat yang terintegrasi dengan ROS, untuk mensimulasikan robot. Dipelajari cara membuat model robot dalam format URDF (Unified Robot Description Format), menambahkan sensor dan aktuator, serta mengontrol robot dalam lingkungan simulasi Gazebo. Contoh diberikan untuk mensimulasikan lengan robotik dan robot beroda, termasuk cara mengumpulkan dan menganalisis data dari simulasi.

Penulis juga menjelaskan bagaimana mengintegrasikan plugin 'gazebo\_ros\_control' untuk mengendalikan robot dalam Gazebo menggunakan ROS. Dipelajari cara membuat lingkungan simulasi yang realistik, menambahkan objek dan rintangan, serta menjalankan skenario simulasi untuk menguji algoritma kontrol dan navigasi robot. Bab ini memberikan landasan kuat untuk menggunakan simulasi dalam pengembangan dan pengujian aplikasi robotika.

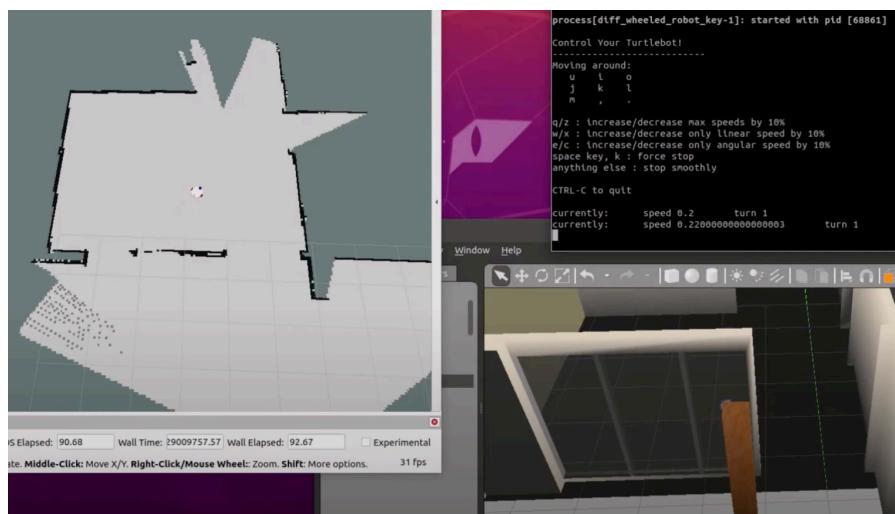
## 5. Bab 5: Simulating Robots Using ROS, CoppeliaSim, and Webots



Bab ini memperkenalkan dua simulator tambahan: CoppeliaSim (sebelumnya V-REP) dan Webots, serta cara menggunakannya dengan ROS. Dipelajari cara mengatur dan menjalankan simulasi di kedua platform ini, serta memahami bagaimana ROS dapat digunakan untuk mengendalikan simulasi di CoppeliaSim dan Webots. Contoh simulasi robotik termasuk manipulasi lengan robotik dan navigasi robot beroda.

Penulis memberikan panduan langkah demi langkah untuk mengintegrasikan ROS dengan kedua simulator, termasuk cara menulis node ROS untuk mengontrol robot dalam simulasi. Dipelajari cara menambahkan sensor dan aktuator, serta membuat skenario simulasi yang kompleks untuk menguji dan memvalidasi algoritma robotika.

## 6. Bab 6: Using the ROS MoveIt! and Navigation Stack



Bab ini

fokus pada MoveIt!, framework untuk perencanaan gerakan (motion planning) yang digunakan dalam robotika. Dipelajari cara mengkonfigurasi dan menggunakan MoveIt! untuk perencanaan gerakan robotik yang kompleks, termasuk manipulasi lengan robotik dan navigasi. Bab ini mencakup konsep-konsep seperti move\_group, planning scene, dan collision checking, serta bagaimana menggunakan Setup Assistant untuk menghasilkan paket konfigurasi MoveIt!.

Selain MoveIt!, bab ini juga membahas penggunaan Navigation Stack ROS untuk mencapai navigasi otonom. Dipelajari cara mengimplementasikan SLAM (Simultaneous Localization and Mapping) untuk pemetaan lingkungan, serta menggunakan algoritma AMCL (Adaptive Monte Carlo Localization) untuk lokalisasi

**TUGAS UAS ROBOTIKA**  
Alfian Mohamad Firdaus | 1103204002

robot. Contoh praktis diberikan untuk menunjukkan cara menggabungkan berbagai komponen ini dalam aplikasi navigasi robotik yang berfungsi penuh.