Soal Nomor 1

a. ROS (Robot Operating System) adalah sebuah *middleware* yang digunakan untuk aplikasi-aplikasi robotik. Fungsi utama ROS adalah sebagai sarana komunikasi antar komponen pada suatu robot. Tujuan adanya ROS adalah untuk membuat suatu standar agar para pengembang robot tidak perlu untuk membuat program yang sama berulang kali. Adanya ROS ini sangat memudahkan para pengembang untuk membuat suatu robot karena mereka bisa membuat bagian-bagian robot tersebut satu-satu kemudian mengintegrasikannya dengan mengguanakan ROS. Contohnya Ketika ingin membuat robot pembuang sampah, pengembang bisa membuat tiga bagian terpisah, yaitu bagian untuk melihat sekitar, bagian untuk memproses hasil penglihatan, dan bagian untuk bergerak ke tempat adanya sampah kemudian mengomunikasikan antara ketiga bagian tersebut menggunakan ROS.

b. Jika kita membandingkan antara ROS1 dan ROS2 ada 7 perbedaan utama yang membedakan keduanya, yaitu
   1. ROS1 memiliki middleware nya sendiri, sedangkan ROS2 menggunakan middleware dari pihak ketiga yaitu dds karena telah terbukti stabil dan agar tidak perlu tambahan Upaya untuk pemeliharaan middleware nya
   2. ROS1 menggunakan ROS master untuk memfasilitasi komunikasi antar node, sedangkan pada ROS2 node-node bisa saling berkomunikasi secara langsung
   3. ROS1 tidak memiliki *shared implementation* untuk c++ dan python, sedangkan ROS2 memiliki *shared implementation* untuk c++ dan python
   4. ROS1 tidak bisa mengoperasikan beberapa node dalam satu proses, sedangkan ROS2 bisa
   5. Pada ROS1 *action services* berada pada *library* yang berbeda, sedangkan pada ROS2 semuanya terintegrasikan dalam satu *library* yang sama
   6. ROS1 tidak kompatibel untuk Windows, sedangkan ROS2 kompatibel
   7. ROS1 menggunakan xml sebagai basis *launch infrastructure*, sedangkan ROS2 menggunakan python

   Dari 7 perbedaan itu jelas bahwasanya ROS2 jauh lebih baik daripada ROS1. Kemudian keuntungan lainnya dari ROS2 adalah karena Sekarang hampir semua projek robotik *open source* telah berpindah ke ROS2 sehingga pengembangan hanya akan dilanjutkan pada ROS2 dan ROS1 hanya akan berhenti pada keadaan saat ini saja.

c. Simulasi robotic sangat penting dalam pengembangan suatu robot. Sebelum prototype robot dibuat di dunia nyata, robot harus dites dulu sebelumnya dengan simulasi. Mengetes robot dalam simulasi memiliki sangat banyak keuntungan terutama dalam efisiensi waktu dan biaya. Sebagai contoh jika suatu Perusahaan ingin membuat suatu robot konstruksi berukuran besar, maka perusahaan tersebut bisa mengetes dulu robot tersebut di dalam dunia simulasi sehingga hanya akan menggunakan biaya yang sangat sedikit dan apabila ada kesalahan, bisa diperbaiki terlebih dahulu agar tidak menimbulkan kecelakaan.

d. Gazebo merupakan *platform* simulasi yang umum digunakan untuk menyimulasikan suatu robot. Gazebo dapat membuat sebuah *environment* yang mirip sekali dengan dunia nyata, seperti gravitasi, gaya gesek, dan lain-lain. Oleh karena itulah gazebo sangat bagus untuk digunakan sebagai dunia simulasi. Gazebo juga bisa diintegrasikan secara langsung dengan ROS menggunakan suatu paket Bernama gazebo_ros, paket ini memungkinkan robot yang berada di gazebo untuk dikendalikan menggunakan ros. Secara simpel, Langkah-langkah untuk mengintegrasikan ros dengan gazebo, yaitu pertama, mengimpor model robot ke dalam simulasi, kedua, mengendalikan robot menggunakan node yang ada di ros atau menggunakan plugin gazebo ros, dan ketiga, memvisualisasikan data menggunakan Rviz.

e. Robot melakukan navigasi dalam dunia simulasi maupun dunia nyata dengan menggunakan alat-alat serta metode-metode tertentu. Pertama-tama yang dilakukan robot adalah *localization* (mengetahui Lokasi robot dalam suatu ruang). Kemudian robot melihat lingkungan sekitar menggunakan kamera untuk dan menghasilkan *mapping* dari lingkungan tersebut. Selanjutnya robot menentukan jalan mana yang akan dipilih untuk mencapai tujuan dan halangan apa yang harus dihindari yang ada pada jalan tersebut. Setelah rancangan pergerakan robot tersebut selesai, maka robot akan bergerak menuju tujuannya dan Kembali kepada posisi awal robot (bisa berbeda tergantung program yang diimplementasikan dalam robot tersebut).

f. TF ata Transform dalam konteks ROS mudahnya adalah hubungan antara dua atau lebih bagian system. Contoh dari transform adalah dalam pergerakan sendi-sendi yang ada pada robot yang diintegrasikan dengan bagian sendi lain serta *map*, pengintegrasian ini memungkinkan robot untuk mengetahui lingkungan sekitar serta bagian lain dirinya sendiri agar tidak menimbulkan *crash* atau tabrakan saat melakukan pergerakan. Ada banyak contoh aplikasi yang bisa dilakukan dengan transform, di antaranya: penghindaran halangan yang ada di lingkungan, menyampaikan Lokasi robot berdasarkan suatu parameter statis, mengonversikan data sensor dari suatu bagian ke bagian lain, dan lain lain.