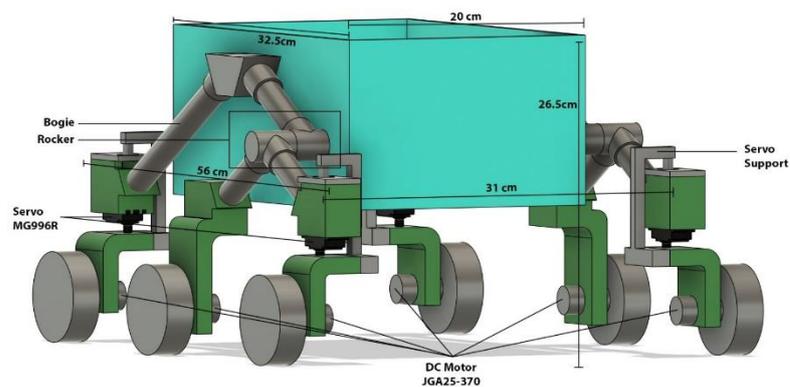


## BAB II

### KONSEP DESAIN & SPESIFIKASI SISTEM

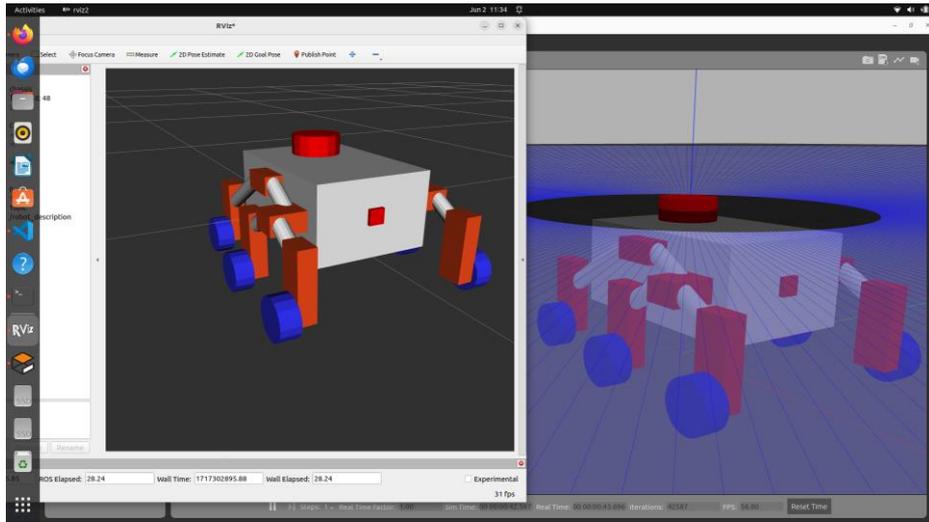
#### 2.1 Konsep Desain Sistem

Konsep sistem *rover* ini adalah robot dapat bergerak secara otonom dalam melakukan pemantauan *indoor* dan *outdoor*. Robot tersebut menggunakan sensor lidar untuk mendeteksi keberadaan objek di sekitar dan menggunakan SLAM untuk melakukan lokalisasi dan *mapping*. *Rover* menggunakan *depth camera* untuk mengambil gambar dan kedalaman objek di depannya. Dalam pengembangannya, *rover* menggunakan 6 roda untuk memudahkan robot saat bergerak dan suspensi untuk pergerakan robot yang lebih stabil dan lebih mudah menghindari rintangan. Pada sistem suspensi terdapat *rocker-bogie* yang berfungsi sebagai *pivot* untuk pergerakan robot yang lebih stabil dan dapat bergerak pada permukaan yang tidak rata seperti yang ditunjukkan pada Gambar 2.1.



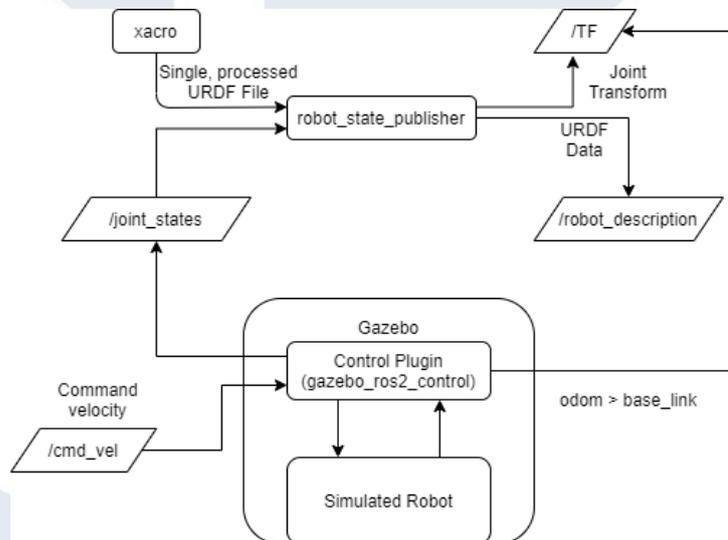
Gambar 2.1 - Konsep Desain 3D Rover

Berdasarkan desain konsep pada Gambar 2.1, *rover* dikembangkan melalui simulasi menggunakan ROS 2 melalui *file-file* xacro yang digabungkan oleh URDF. Kemudian URDF tersebut akan di *publish* oleh ROS 2 dan dapat ditampilkan di dalam gazebo. Pada implementasi simulasi ini robot bergerak secara otonom menggunakan NAV2 dan bergerak dari titik *start* ke titik tujuan menggunakan algoritma A\* yang terdapat di dalam NAV2. Implementasi ini dilakukan menggunakan sekumpulan *file* dalam format URDF, yang bersama-sama mendeskripsikan robot tersebut. *File-file* tersebut diproses oleh xacro dan digabungkan menjadi satu URDF. Kemudian selanjutnya *robot\_state\_publisher* yang membuat data menjadi tersedia mengenai */robot\_description* dari topik tersebut dan menampilkan bentuk transformasinya seperti yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 - Transformasi Simulasi Model Robot Rover di Gazebo dan Rviz 2

Keseluruhan dari sistem simulasi robot berinteraksi dengan menggunakan publikasi deskripsi robot. Robot ditampilkan oleh `/robot_description`, dan status gabungan tersebut dilakukan oleh *plugin* kontrol. *Plugin* tersebut menyiarkan sebuah *transform* dari *frame* baru (*odom*) ke *base\_link* yang memungkinkan kode lain mengetahui perkiraan posisi robot pada saat ini. *Odom* merupakan bentuk *transform* yang menunjukkan asal dunia dan posisi awal robot. Hal tersebut ditunjukkan pada Gambar 2.3. Untuk memudahkan proses implementasi, semua program yang telah dibuat digabungkan ke dalam *file launch* sehingga mempercepat proses *launch package* seperti yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 - Keseluruhan Sistem Robot pada ROS 2 dan Gazebo

Pembuatan model robot di dalam xacro menggunakan nilai massa, koordinat xyz, radius, rpy dan *length*. Pembuatan model *chassis* dan *mount* menggunakan balok yang terdiri dari nilai panjang, lebar, dan tinggi. Kemudian pada pembuatan suspensi menggunakan tabung dan memasukkan parameter radius dan *length*. Kemudian pada roda menggunakan lingkaran dengan parameter radius

dan *length*. Di dalam xacro terdapat *origin xyz* yang menentukan koordinat dari *link* tersebut. Dengan menggunakan parameter *rpy* memudahkan pengguna dalam menentukan rotasi objek. Pada pembuatan model robot menggunakan nilai momen inersia karena mempengaruhi bagaimana objek merespon terhadap gaya dan momen berdasarkan persamaan (2.1)[27].

$$I = \frac{1}{2}mr^2 \dots\dots\dots (2.1)$$

Sumber : *Fundamentals of Physics* – Hal. 255

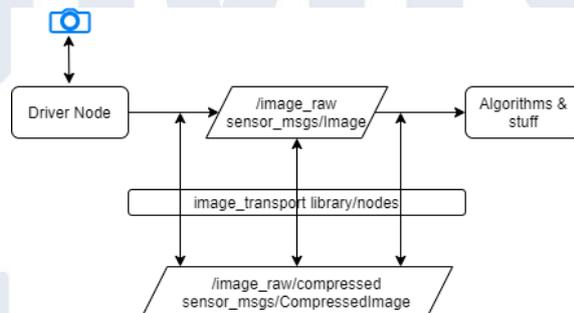
Persamaan momen inersia :  
 I merupakan momen inersia.  
 m merupakan massa benda.  
 r merupakan jari-jari .

*Rover* menggunakan subsistem navigasi yang berfungsi untuk robot dapat mengenali lingkungan di sekitarnya. Subsistem navigasi pada robot ini menggunakan *depth camera* dan *lidar* sebagai sensor. Kamera dapat mengambil gambar, warna dan kedalaman untuk ditampilkan di dalam RViz 2 berdasarkan konsep sistem seperti yang ditunjukkan pada Gambar 2.4.



Gambar 2.4 - Konsep Sistem *Depth Camera* dalam Mengambil Gambar di Simulasi

Kamera memiliki *node driver* yang dirancang untuk berkomunikasi dengan *hardware* kamera dan mempublikasikan aliran data. *Node driver* bekerja mengontrol resolusi dan *frame rate*. Aliran data kamera yang diambil oleh *node driver* akan di *publish* ke *topic* tipe *sensor\_msgs/Image*. Algoritma dari kamera berfungsi sebagai *image message*, dan selama kamera mempunyai *driver* maka algoritmanya akan berfungsi. *Topic* yang belum diproses oleh *image* dan dipublikasikan langsung oleh kamera dinamakan */image\_raw*. Jika terdapat *image* dengan versi yang terkompresi, maka akan ditambahkan */compressed* ke dalam *topic* */image\_raw/compressed*. Untuk membantu dalam proses mengompresi gambar, dapat menggunakan *tools image\_transport*. *Image message* mempunyai *Header* yang berisi waktu pengukuran gambar dan koordinat *frame* kamera. Hal ini ditunjukkan pada Gambar 2.5.



Gambar 2.5 - *Node Driver* Mengambil Data dari Kamera

Mobile robot dapat melakukan *mapping* dan *localization* menggunakan sensor lidar. Ketika simulasi robot, sensor lidar mengambil data berupa *point clouds* objek di sekitar robot kemudian menjadikannya *topic laserscan*. *Topic* ini akan ditampilkan di dalam RViz 2 kemudian pemetaan lingkungan dapat dibuat di dalam RViz 2 seperti yang ditunjukkan konsep sistem lidar pada Gambar 2.6.



Gambar 2.6 - Konsep Sistem *Laserscan* dari Sensor Lidar

Pada sistem pergerakan robot, robot bergerak secara otonom dengan cara membuat program NAV2 yang di dalamnya terdapat algoritma A\*. Algoritma A\* pada sistem NAV2 memudahkan robot dalam melakukan pergerakan dari titik *start* ke titik tujuan seperti yang ditunjukkan pada konsep sistem *path planning* pada Gambar 2.7. Berdasarkan Gambar 2.7, parameter NAV2 akan menampilkan *topic /global/map* yang membantu pengguna dalam membaca gambar hasil *mapping*. Parameter tersebut dijalankan di dalam RViz 2 dan pengguna juga dapat memilih titik tujuan robot menggunakan *twist\_mux* yang diprogram ke dalam */cmd\_vel*. Proses ini dilakukan di dalam *software* RViz 2.



Gambar 2.7 - Konsep Sistem *Path Planning* Algoritma A\*

## 2.2 Spesifikasi Sistem

### 2.2.1 Subsistem Navigasi

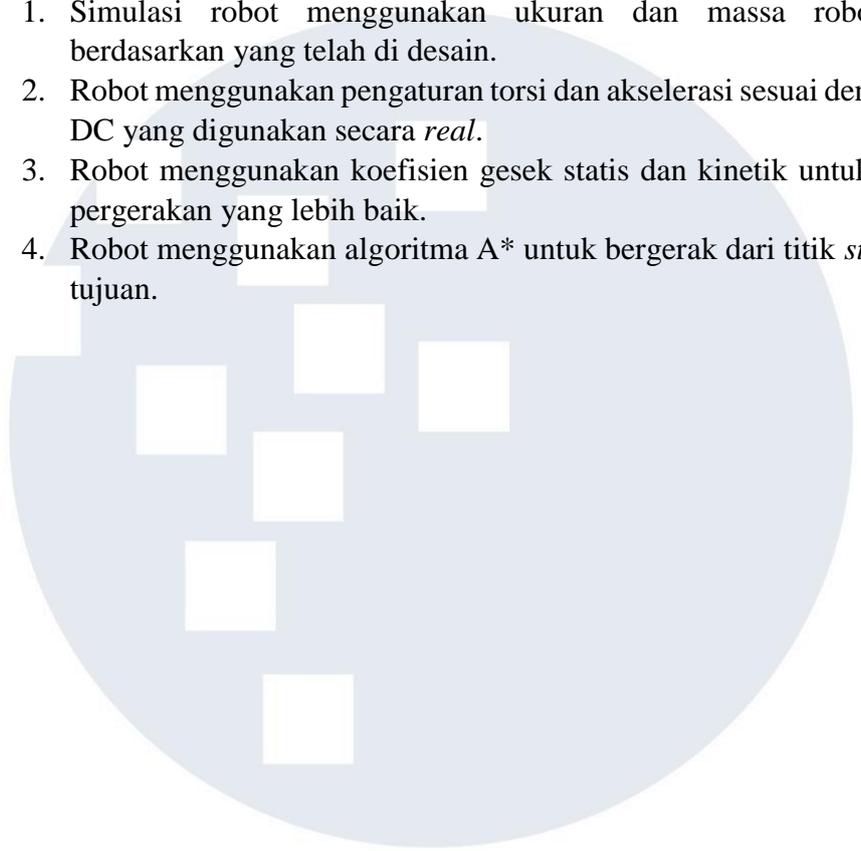
Spesifikasi yang ingin dicapai pada subsistem navigasi pada simulasi ini yaitu.

1. Hasil gambar, warna, dan kedalaman dapat ditampilkan di dalam RViz 2.
2. Data *point clouds* dari *topic laserscan* dapat ditampilkan di dalam *software* RViz 2.
3. Robot menggunakan metode SLAM *toolbox* dan ditampilkan di dalam RViz 2.
4. Robot dapat bergerak dari titik *start* ke titik tujuan.

### 2.3 Metode Verifikasi Spesifikasi

Untuk memastikan bahwa simulasi robot sesuai dengan spesifikasi yang diharapkan, terdapat beberapa metode yang dilakukan untuk mengembangkan simulasi.

1. Simulasi robot menggunakan ukuran dan massa robot aslinya berdasarkan yang telah di desain.
2. Robot menggunakan pengaturan torsi dan akselerasi sesuai dengan motor DC yang digunakan secara *real*.
3. Robot menggunakan koefisien gesek statis dan kinetik untuk performa pergerakan yang lebih baik.
4. Robot menggunakan algoritma A\* untuk bergerak dari titik *start* ke titik tujuan.



UMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA