



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**PENGEMBANGAN LINGKUNGAN SIMULASI
UNTUK PENGUJIAN *SOCIALLY ASSISTIVE*
ROBOTS MENGGUNAKAN ROS 2 DAN GAZEBO**

Muhammad Alfi Maulana Fikri
NRP 0721 17 4000 0009

Dosen Pembimbing
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. I Ketut Eddy Purnama, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**PENGEMBANGAN LINGKUNGAN SIMULASI
UNTUK PENGUJIAN *SOCIALLY ASSISTIVE*
ROBOTS MENGGUNAKAN ROS 2 DAN GAZEBO**

Muhammad Alfi Maulana Fikri
NRP 0721 17 4000 0009

Dosen Pembimbing
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. I Ketut Eddy Purnama, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

[Halaman ini sengaja dikosongkan]



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC184801

**DEVELOPMENT OF SIMULATION
ENVIRONMENT FOR SOCIALLY ASSISTIVE
ROBOTS TESTING USING ROS 2 AND GAZEBO**

Muhammad Alfi Maulana Fikri
NRP 0721 17 4000 0009

Advisors

Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. I Ketut Eddy Purnama, S.T., M.T.

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical and Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2021

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi buku Tugas Akhir dengan judul “**Pengembangan Lingkungan Simulasi untuk Pengujian *Socially Assistive Robots* Menggunakan ROS 2 dan Gazebo**” adalah benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2021

Muhammad Alf Maulana Fikri
0721 17 4000 0009

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENGEMBANGAN LINGKUNGAN SIMULASI UNTUK PENGUJIAN *SOCIALLY ASSISTIVE* *ROBOTS* MENGGUNAKAN ROS 2 DAN GAZEBO

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Muhammad Alfi Maulana Fikri (NRP. 0721 17 4000 0009)

Tanggal Ujian : 17 Juni 2021

Periode Wisuda : September 2021

Disetujui Oleh:

Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng. (Pembimbing I)
NIP: 9580916 198601 1 001

Dr. I Ketut Eddy Purnama, S.T., M.T. (Pembimbing II)
NIP: 9690730 199512 1 001

..... (Penguji I)
NIP:

..... (Penguji II)
NIP:

..... (Penguji III)
NIP:

Mengetahui,
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
NIP. 9700313 199512 1 001

[Halaman ini sengaja dikosongkan]

ABSTRAK

Nama : Muhammad Alfı Maulana Fikri
Judul : Pengembangan Lingkungan Simulasi untuk Pengujian *Socially Assistive Robots* Menggunakan ROS 2 dan Gazebo
Pembimbing : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
2. Dr. I Ketut Eddy Purnama, S.T., M.T.

Pada penelitian ini kami mengajukan lingkungan simulasi yang dikembangkan menggunakan ROS 2 dan Gazebo untuk pengujian *socially assistive robots* (SARs). Di dalam lingkungan simulasi ini, model robot yang digunakan akan diujikan dengan model pengguna serta model-model objek lain secara virtual. Untuk mempermudah pemindahan program dari simulasi ke robot fisik, kontroler robot akan dikembangkan secara terpisah dari lingkungan simulasi yang mana ketika pengujian, keduanya akan saling terhubung menggunakan sistem komunikasi antar proses yang ada di ROS 2. Diharapkan, lingkungan simulasi yang dibuat dapat membantu pengujian SARs dengan meminimalisir resiko, mengurangi biaya, dan menghemat waktu jika dibandingkan dengan melakukan pengujian secara langsung menggunakan robot fisik.

Kata Kunci: Simulasi, *Assistive Robotics*, ROS2, Gazebo.

[Halaman ini sengaja dikosongkan]

ABSTRACT

Name : Muhammad Alfi Maulana Fikri
Title : *Development of Simulation Environment for Socially Assistive Robots Testing Using ROS 2 and Gazebo*
Advisors : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
2. Dr. I Ketut Eddy Purnama, S.T., M.T.

In this study, we propose a simulation environment developed using ROS 2 and Gazebo for socially assistive robots (SARs) testing. In this simulation environment, the robot model used will be tested with the user model and other virtual object models. To make it easier to transfer the program from simulation to physical robot, the robot controller will be developed separately from the simulation environment in which when testing, both will be connected to each other using ROS 2 interprocess communication system. It is expected that the simulation environment created can assist the testing of SARs by minimizing risk, reducing costs, and saving time when compared to conducting direct testing using physical robots.

Keywords: *Simulation, Assistive Robotics, ROS 2, Gazebo.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul “**Pengembangan Lingkungan Simulasi untuk Pengujian *Socially Assistive Robots* Menggunakan ROS 2 dan Gazebo**”.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Bapak, Ibu dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian penelitian ini.
2. Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. selaku Kepala Departemen Teknik Komputer - FTEIC ITS.
3. Bapak Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng., Bapak Dr. I Ketut Eddy Purnama, S.T., M.T., dan Bapak Muhtadin ST., MT. atas arahan dan bimbingan selama pengerjaan penelitian tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh rekan-rekan ICHIRO-ITS, B201 crew, dan penghuni rumah anak TK.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juni 2021

Muhammad Alfi Maulana Fikri

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|---|-------------|
| ABSTRAK | i |
| ABSTRACT | iii |
| KATA PENGANTAR | v |
| DAFTAR ISI | vii |
| DAFTAR GAMBAR | xi |
| DAFTAR KODE | xiii |
| DAFTAR TABEL | xv |
| 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Permasalahan | 2 |
| 1.3 Tujuan | 2 |
| 1.4 Batasan Masalah | 2 |
| 1.5 Sistematika Penulisan | 3 |
| 2 TINJAUAN PUSTAKA | 5 |
| 2.1 <i>Assistive Robotics</i> | 5 |
| 2.1.1 <i>Socially Assistive Robots</i> (SARs) | 6 |
| 2.2 Gazebo | 6 |
| 2.2.1 Gazebo Model | 7 |
| 2.2.2 Gazebo Plugins | 8 |

| | | |
|----------|--|-----------|
| 2.3 | Simulation Description Format (SDFormat) | 8 |
| 2.4 | Robot Operating System 2 (ROS 2) | 9 |
| 2.4.1 | ROS 2 Middleware Implementations (RMW) | 10 |
| 2.4.2 | ROS 2 Client Libraries (RCL) | 11 |
| 2.4.3 | ROS 2 Node | 12 |
| 2.4.4 | ROS 2 Topic | 12 |
| 2.4.5 | ROS 2 Service | 13 |
| 2.4.6 | ROS 2 Command Line Interface | 14 |
| 2.4.7 | RQt | 14 |
| 2.4.8 | Rviz2 | 15 |
| 2.5 | <i>Holonomic Robot</i> | 16 |
| 2.6 | <i>Inertial Measurement Unit (IMU)</i> | 16 |
| 2.7 | <i>Smart Assistive Posture Device</i> | 17 |
| 2.8 | <i>Depth Camera</i> | 18 |
| 2.8.1 | Proyeksi Kamera | 18 |
| 2.8.2 | <i>Point Cloud</i> | 19 |
| 2.9 | Kinect V2 | 20 |
| 2.9.1 | Freenect 2 Library (libfreenect2) | 20 |
| 2.10 | MediaPipe | 21 |
| 2.10.1 | BlazePose | 22 |
| 2.11 | SLAM | 22 |
| 2.11.1 | RTAB-Map | 23 |
| 3 | DESAIN DAN IMPLEMENTASI SISTEM | 25 |
| 3.1 | Pengembangan Model Robot | 25 |
| 3.1.1 | Desain Robot <i>Dienen</i> | 25 |
| 3.1.2 | Prototipe Robot <i>Dienen</i> | 27 |
| 3.1.3 | Struktur SDFormat Robot | 28 |

| | | |
|----------|---|-----------|
| 3.2 | Pengembangan Model Pengguna | 31 |
| 3.2.1 | Struktur SDFormat Pengguna | 32 |
| 3.3 | Pengembangan Lingkungan Simulasi | 33 |
| 3.3.1 | Lingkungan Simulasi <i>Outdoor</i> | 33 |
| 3.3.2 | Lingkungan Simulasi <i>Indoor</i> | 35 |
| 3.4 | Integrasi <i>Plugin</i> Untuk Simulasi | 36 |
| 3.4.1 | <i>Navigation Plugin</i> | 37 |
| 3.4.2 | <i>Camera Plugin</i> | 39 |
| 3.4.3 | <i>Depth Camera Plugin</i> | 39 |
| 3.4.4 | <i>Legs Plugin</i> | 40 |
| 3.5 | Pengembangan <i>Behavior Node</i> | 43 |
| 3.6 | Integrasi Sistem pada <i>Real Robot</i> | 43 |
| 4 | PENGUJIAN DAN ANALISIS | 45 |
| 4.1 | Pengujian Gerakan | 46 |
| 4.1.1 | Pengujian Gerakan Linier dan Estimasi Posisi di Simulasi | 46 |
| 4.1.2 | Pengujian Gerakan Linier dan Estimasi Posisi pada <i>Real Robot</i> | 49 |
| 4.1.3 | Pengujian Gerakan Putar dan Estimasi Orientasi di Simulasi | 51 |
| 4.1.4 | Pengujian Gerakan Putar dan Estimasi Orientasi pada <i>Real Robot</i> | 53 |
| 4.2 | Pengujian Citra Kamera | 54 |
| 4.2.1 | Pengujian Pengiriman Citra Kamera di Simulasi | 54 |
| 4.2.2 | Pengujian Pengiriman Citra Kamera pada <i>Real Robot</i> | 55 |
| 4.2.3 | Pengujian Pengiriman Citra Kamera Antar-perangkat di Simulasi | 57 |

| | | |
|----------|--|-----------|
| 4.2.4 | Pengujian Pengiriman Citra Kamera Antar-perangkat pada <i>Real Robot</i> | 58 |
| 4.3 | Pengujian <i>Depth Camera</i> | 59 |
| 4.3.1 | Pengujian Citra <i>Depth Camera</i> di Simulasi | 60 |
| 4.3.2 | Pengujian Citra <i>Depth Camera</i> pada <i>Real Robot</i> | 61 |
| 4.3.3 | Pengujian <i>Point Cloud</i> dari Citra <i>Depth Camera</i> di Simulasi | 63 |
| 4.3.4 | Pengujian <i>Point Cloud</i> dari Citra <i>Depth Camera</i> di Robot | 64 |
| 4.4 | Pengujian Klasifikasi Pose | 64 |
| 4.4.1 | Pengujian Klasifikasi Pose di Simulasi | 65 |
| 4.4.2 | Pengujian Klasifikasi Pose di Robot | 66 |
| 4.5 | Pengujian SLAM | 66 |
| 4.5.1 | Pengujian SLAM di Simulasi | 67 |
| 4.5.2 | Pengujian SLAM di Robot | 68 |
| 5 | PENUTUP | 69 |
| 5.1 | Kesimpulan | 69 |
| 5.2 | Saran | 69 |
| | DAFTAR PUSTAKA | 71 |

DAFTAR GAMBAR

| | | |
|-----|--|----|
| 2.1 | Pengategorian <i>assistive robots</i> (Heerink et al. [1]). . . | 6 |
| 2.2 | Diagram komunikasi antar node pada ROS 2 [2]. . . | 10 |
| 3.1 | Desain dan diagram komponen dari robot <i>Dienen</i> . . . | 26 |
| 3.2 | Diagram komponen dari prototipe robot <i>Dienen</i> . . . | 27 |
| 3.3 | Tampilan lingkungan simulasi <i>outdoor</i> pada Gazebo. . . | 34 |
| 3.4 | Tampilan lingkungan simulasi <i>indoor</i> pada Gazebo. . . | 36 |
| 3.5 | Diagram integrasi <i>plugin</i> untuk simulasi. | 37 |
| 3.6 | Diagram integrasi <i>plugin</i> untuk model pengguna di simulasi. | 41 |
| 3.7 | Diagram integrasi sistem pada <i>real robot</i> | 43 |
| 4.1 | Relasi antar- <i>node</i> dari pengujian gerakan linier dan estimasi posisi di simulasi. | 46 |
| 4.2 | Grafik estimasi posisi dari gerakan linier di simulasi. | 48 |
| 4.3 | Relasi antar- <i>node</i> dari pengujian gerakan linier dan estimasi posisi pada <i>real robot</i> | 49 |
| 4.4 | Grafik estimasi posisi dari gerakan linier pada <i>real robot</i> | 51 |
| 4.5 | Grafik estimasi orientasi dari gerakan putar di simulasi. | 52 |
| 4.6 | Grafik estimasi orientasi dari gerakan putar pada <i>real robot</i> | 54 |
| 4.7 | Relasi antar- <i>node</i> dari pengujian pengiriman citra kamera di simulasi. | 55 |
| 4.8 | Grafik <i>delay</i> dan frekuensi dari pengiriman citra di simulasi. | 56 |

| | | |
|------|--|----|
| 4.9 | Relasi antar- <i>node</i> dari pengujian pengiriman citra kamera pada <i>real robot</i> | 57 |
| 4.10 | Grafik <i>delay</i> dan frekuensi dari pengiriman citra pada <i>real robot</i> | 58 |
| 4.11 | Grafik <i>delay</i> dan frekuensi dari pengiriman citra antar-perangkat di simulasi. | 59 |
| 4.12 | Grafik <i>delay</i> dan frekuensi dari pengiriman citra antar-perangkat pada <i>real robot</i> | 60 |
| 4.13 | Relasi antar- <i>node</i> dari pengujian citra <i>depth camera</i> di simulasi. | 61 |
| 4.14 | Perbandingan hasil tangkapan citra berwarna dan citra kedalaman di simulasi. | 62 |
| 4.15 | Relasi antar- <i>node</i> dari pengujian citra <i>depth camera</i> pada <i>real robot</i> | 62 |
| 4.16 | Perbandingan hasil tangkapan citra berwarna dan citra kedalaman pada <i>real robot</i> | 63 |

DAFTAR KODE

| | | |
|------|---|----|
| 3.1 | <i>Link element</i> untuk bagian kepala robot. | 29 |
| 3.2 | <i>Joint element</i> yang menghubungkan bagian pundak kiri dan bagian badan atas robot. | 29 |
| 3.3 | <i>Joint element</i> yang menghubungkan bagian badan atas dan bagian badan bawah robot. | 30 |
| 3.4 | <i>Sensor element</i> dari sensor kamera. | 30 |
| 3.5 | <i>Sensor element</i> dari sensor <i>depth camera</i> | 31 |
| 3.6 | Struktur SDFFormat dari model pengguna. | 32 |
| 3.7 | Struktur SDFFormat dari lingkungan simulasi <i>outdoor</i> | 34 |
| 3.8 | Struktur SDFFormat dari lingkungan simulasi <i>indoor</i> | 35 |
| 3.9 | <i>Class</i> dari <i>navigation plugin</i> | 37 |
| 3.10 | Integrasi <i>navigation plugin</i> pada model robot. | 38 |
| 3.11 | Integrasi <i>camera plugin</i> pada model robot. | 39 |
| 3.12 | Integrasi <i>depth camera plugin</i> pada model robot. | 40 |
| 3.13 | <i>Class</i> dari <i>legs plugin</i> | 41 |
| 3.14 | Integrasi <i>legs plugin</i> pada model pengguna. | 42 |

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

| | | |
|------|--|----|
| 4.1 | Spesifikasi komputer untuk menjalankan simulator. . | 45 |
| 4.2 | Spesifikasi komputer yang ada pada prototipe robot. | 45 |
| 4.3 | Hasil estimasi posisi dari gerakan linier di simulasi selama 10 detik. | 47 |
| 4.4 | Hasil estimasi posisi dari gerakan linier pada <i>real robot</i> selama 3 detik. | 50 |
| 4.5 | Hasil estimasi orientasi dari gerakan putar di simulasi selama 10 detik. | 52 |
| 4.6 | Hasil estimasi orientasi dari gerakan putar pada <i>real robot</i> selama 3 detik. | 53 |
| 4.7 | Hasil <i>delay</i> dan frekuensi dari pengiriman citra kamera di simulasi. | 55 |
| 4.8 | Hasil <i>delay</i> dan frekuensi dari pengiriman citra kamera pada <i>real robot</i> | 57 |
| 4.9 | Hasil <i>delay</i> dan frekuensi dari pengiriman citra kamera antar-perangkat di simulasi. | 58 |
| 4.10 | Hasil <i>delay</i> dan frekuensi dari pengiriman citra kamera antar-perangkat pada <i>real robot</i> | 59 |

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar Belakang

Selama beberapa tahun terakhir, robot telah mengalami perkembangan yang signifikan dari robot beroda untuk edukasi [3] hingga robot manipulator untuk skala industri [4]. Salah satu bentuk perkembangan lain dari robot tersebut adalah *socially assistive robots* (SARs). SARs merupakan jenis robot dalam bidang *socially assistive robotics* yang menggabungkan aspek yang ada pada *assistive robotics* dan *socially interactive robotics* sehingga menjadikan SARs sebagai robot yang mampu memberikan bantuan kepada pengguna dalam bentuk interaksi sosial [5].

Namun, karena sifat dari SARs yang melibatkan interaksi langsung dengan pengguna, maka pengujian dari robot akan menjadi sulit dan beresiko bagi pengguna yang ikut terlibat dalam pengujian tersebut [6]. Salah satu solusi untuk mengatasi masalah tersebut adalah dengan melakukan pengujian secara virtual melalui simulasi. Selain bisa meminimalisir resiko, penggunaan simulasi juga bisa mengurangi biaya yang dibutuhkan dan menghemat waktu pengujian selama pengembangan robot tersebut [7].

Hingga saat ini sudah ada beberapa simulator yang bisa digunakan untuk menjalankan simulasi robot seperti Webots [8], Gazebo [9], V-REP [10], OpenAI Gym [11], dan lain sebagainya. Namun, simulator-simulator tersebut hanyalah platform yang secara umum digunakan untuk membantu pengembangan robot melalui simulasi virtual. Sedangkan pengembangan dari lingkungan simulasi dan kontroler robot untuk simulasi tersebut harus dibuat sendiri oleh pengembang robot.

Untuk itu, pada penelitian ini kami mengajukan penelitian terkait pengembangan lingkungan simulasi untuk pengujian SARs menggunakan ROS 2 dan Gazebo. ROS 2 dan Gazebo sendiri dipilih karena tersedianya banyak library yang dapat membantu pengembangan maupun pengujian robot, terutama untuk simulasi. Selain itu, dengan adanya ROS 2, kontroler robot yang diuji melalui simulasi bisa dengan mudah dipindahkan ke robot fisik untuk diuji secara langsung pada pengguna [7].

1.2 Permasalahan

Dari permasalahan tersebut, maka masalah yang dapat diambil adalah Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna.

1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.
2. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

1.4 Batasan Masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut di antaranya adalah:

1. Mempermudah Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.
2. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.
3. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan, dan metodologi penelitian.

2. BAB II Tinjauan Pustaka

Bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

3. BAB III Desain dan Implementasi Sistem

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang hasil serta analisis yang didapatkan dari pengujian yang dilakukan mulai dari hasil pengujian Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

[Halaman ini sengaja dikosongkan]

BAB II

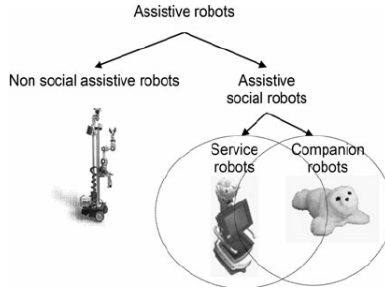
TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 *Assistive Robotics*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Gambar 2.1: Pengategorian *assistive robots* (Heerink et al. [1]).

2.1.1 *Socially Assistive Robots (SARs)*

Socially assistive robots (SARs) merupakan adaptasi dari *assistive technology* yang meliputi keseluruhan sistem robotika yang mampu memberikan bantuan kepada pengguna dalam bentuk interaksi sosial [5]. Heerink et al. [1] mengategorikan riset terhadap SARs menjadi dua kategori berbeda seperti pada Gambar 2.1. Kategori pertama mencakup *service robots* yang menawarkan bantuan fisik dan kognitif dan melakukan tugas sebagai pelayan, sedangkan kategori kedua mencakup *companion robot* yang merupakan robot berjenis pendamping sebagai sahabat dan media untuk terapi.

Lebih lanjut Rich dan Sidner [12], menjelaskan SARs mampu memberikan bantuan kepada pengguna dalam berbagai tingkatan seperti: (a) mendukung kemampuan fungsional dan kognitif pengguna; (b) menawarkan pengguna kesempatan untuk meningkatkan partisipasi sosial dan kesehatan psikologis; (c) menyediakan pemantauan jarak jauh dan berkelanjutan atas status kesehatan pengguna; dan (d) membina pengguna untuk memfasilitasi promosi perilaku sehat dan pencapaian tujuan yang berhubungan dengan kesehatan.

2.2 Gazebo

Gazebo [9] merupakan bagian dari Player Project [13] yang memungkinkan sebuah simulasi robot dan aplikasi sensor bekerja di lingkungan simulasi *indoor* maupun *outdoor* tiga dimensi. Ga-

zebo memiliki arsitektur *client/server* dan model *publish/subscribe* untuk sistem komunikasi antar prosesnya. Setiap objek simulasi di Gazebo dapat diasosiasikan dalam satu maupun lebih kontroler yang akan memproses perintah untuk mengatur dan menentukan keadaan dari suatu objek. Data yang dihasilkan oleh suatu kontroler akan dikirim ke *shared memory* menggunakan *Gazebo interfaces (ifaces)*. Nantinya *ifaces* dari proses-proses lain dapat membaca data tersebut pada *shared memory*, sehingga memungkinkan komunikasi antar proses antara program yang mengontrol robot dan Gazebo, terlepas dari bahasa pemrograman yang digunakan.

2.2.1 Gazebo Model

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.2.2 Gazebo Plugins

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.3 Simulation Description Format (SDFormat)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada

eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

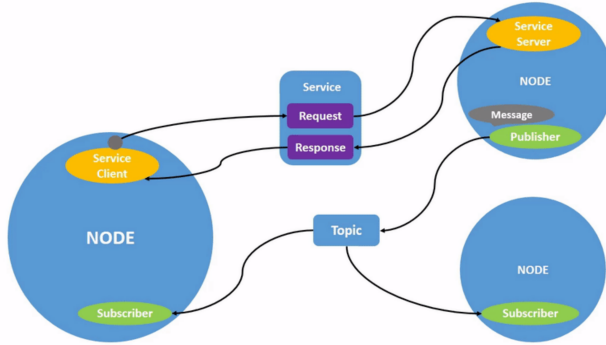
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4 Robot Operating System 2 (ROS 2)

Robot Operating System (ROS) [14] merupakan kumpulan dari *libraries*, *drivers*, dan *tools* yang mempermudah pengembangan sistem pada robot. ROS memiliki *command tool* seperti Linux, sistem komunikasi antar proses, dan berbagai macam *packages* yang berhubungan dengan pengembangan sistem pada robot. Proses yang dieksekusi pada ROS disebut sebagai *node*, komunikasi antar proses yang dimiliki menggunakan model *publish/subscribe*, dan data komunikasi yang dikirimkan disebut sebagai *topic*.

Seperti yang terlihat pada Gambar 2.2, Suatu proses *publisher* mampu mengirimkan satu maupun lebih *topic*, kemudian proses-proses lain yang melakukan *subscribe* pada suatu *topic* bisa memperoleh isi dari *topic* tersebut. Selain itu ada juga *service* yang memiliki fungsi seperti *topic*, hanya saja dilakukan secara dua arah. *Service* ini bekerja menggunakan model *client/server* dimana *service client* akan mengirimkan data permintaan dalam bentuk *request* dan kemudian *service server* akan mengirimkan data balasan dalam bentuk *response*.

Generasi kedua dari Robot Operating System, ROS 2, merupakan kelanjutan dari ROS yang mengusung reliabilitas dan performa untuk penggunaan *real-time* sembari masih mendukung keunggulan yang dimiliki oleh ROS sebelumnya [15]. Untuk memenu-



Gambar 2.2: Diagram komunikasi antar node pada ROS 2 [2].

hi kebutuhan reliabilitas dan performa untuk penggunaan *real-time* tersebut, ROS 2 menggunakan *Data Distribution Service* (DDS) [16] [17], standar industri untuk sistem komunikasi *real-time* dan *end-to-end middleware*, yang menggantikan sistem komunikasi antar proses yang dimiliki ROS sebelumnya.

2.4.1 ROS 2 Middleware Implementations (RMW)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.2 ROS 2 Client Libraries (RCL)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.3 ROS 2 Node

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.4 ROS 2 Topic

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla.

Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.5 ROS 2 Service

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.6 ROS 2 Command Line Interface

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.7 RQt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla.

Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.4.8 Rviz2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.5 *Holonomic Robot*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.6 *Inertial Measurement Unit (IMU)*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada

eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.7 Smart Assistive Posture Device

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam

tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.8 *Depth Camera*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.8.1 **Proyeksi Kamera**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Pha-

sellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.8.2 Point Cloud

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.

Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.9 Kinect V2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.9.1 Freenect 2 Library (libfreenect2)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et ma-

lesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.10 MediaPipe

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at,

lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.10.1 BlazePose

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.11 SLAM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Cu-

rabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.11.1 RTAB-Map

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

BAB III

DESAIN DAN IMPLEMENTASI SISTEM

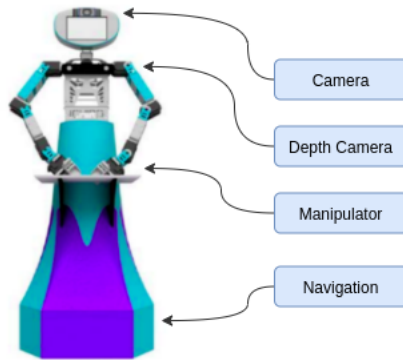
Pada bab ini akan diuraikan desain dan implementasi dari sistem yang telah dibuat. Desain dan implementasi yang akan diuraikan ini dimulai dari perancangan model dan lingkungan yang akan digunakan di simulasi, lalu integrasi ROS 2 pada simulasi sebagai *Gazebo plugins*, kemudian pengembangan program *behavior* untuk pengujian, dan terakhir integrasi sistem yang ada pada *real robot* melalui pengembangan *controller node* sebagai pengganti dari *Gazebo plugins* yang ada di simulasi.

3.1 Pengembangan Model Robot

Pengembangan model robot untuk simulasi dilakukan dengan memperhatikan desain dari robot yang digunakan pada penelitian. Dari desain tersebut, model 3D yang ada perlu dipisah menyesuaikan bagian yang dapat bergerak bebas di robot seperti tangan dan kepala. Terakhir, file SDF format (*simulation description format*) akan dibuat dengan isi yang menyesuaikan sensor, aktuator, dan setiap bagian 3D yang ada pada robot sehingga model tersebut bisa digunakan di simulator Gazebo.

3.1.1 Desain Robot *Dienen*

Robot yang digunakan pada penelitian ini adalah robot *Dienen*, yang merupakan kelanjutan dari robot *IRIS* [18][19] dengan penambahan bagian badan, lengan, dan kepala dari robot *ICHIRO* [20] di bagian atas robot. Desain seperti ini secara umum dikenal sebagai desain *mobile humanoid robot* [21], yang merupakan desain gabungan antara robot *mobile* dan robot *humanoid*. Seperti yang terlihat pada gambar 3.1, robot *Dienen* memiliki beberapa komponen yang bisa dikelompokkan menjadi 4 bagian terpisah, yakni sebuah kamera, *depth camera*, *manipulator*, dan navigasi.



Gambar 3.1: Desain dan diagram komponen dari robot *Dienen*.

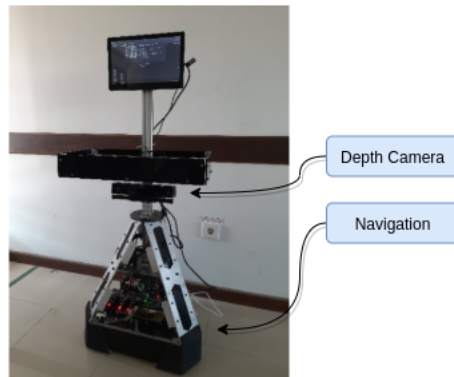
Komponen kamera yang ada pada robot digunakan untuk menangkap citra yang nantinya bisa digunakan untuk melakukan proses visi komputer. Dengan adanya kamera ini, robot diharapkan mampu memperoleh masukan yang bisa berupa deteksi pengguna maupun deteksi objek yang ada di lingkungan.

Komponen *depth camera* yang ada pada robot dapat digunakan untuk menangkap dua macam citra, yakni citra berwarna dan citra kedalaman (*depth image*). Citra gambar yang ditangkap memiliki format RGBA sedangkan citra kedalaman yang ditangkap memiliki format *grayscale* dengan nilai yang semakin gelap menunjukkan jarak yang lebih jauh dari kamera. *Depth Camera* ini nantinya bisa digunakan untuk melakukan pemetaan ruangan sekaligus meningkatkan keakuratan posisi dan orientasi dari robot menggunakan metode SLAM (*simultaneous localization and mapping*).

Komponen *manipulator* yang ada pada robot terdiri dari dua buah lengan dengan lima buah *joints* di setiap lengan. Lengan *manipulator* ini memiliki sebuah *gripper* di ujung lengan yang dapat digunakan untuk menggenggam sebuah objek. Dengan adanya lengan *manipulator* ini, robot diharapkan mampu memberikan tindakan *assistive* seperti membantu untuk membawakan maupun memberikan sebuah objek kepada pengguna.

Komponen navigasi yang ada pada robot merujuk pada komponen yang memungkinkan pergerakan yang ada pada robot. Komponen ini terdiri atas tiga buah *omnidirectional wheels* yang terpasang secara *holonomic* sehingga memungkinkan pergerakan ke segala arah, dua buah sensor *rotary encoder* yang dapat digunakan untuk memperkirakan posisi dari robot, dan sebuah sensor IMU (*inertial measurement unit*) yang dapat digunakan untuk memperkirakan orientasi dari robot. Dengan adanya komponen Navigasi ini, robot diharapkan mampu berpindah tempat secara mudah dari suatu posisi ke posisi yang lain.

3.1.2 Prototipe Robot *Dienen*



Gambar 3.2: Diagram komponen dari prototipe robot *Dienen*.

Prototipe robot *Dienen* ini adalah pengembangan sementara dari robot *Dienen*. Agar sesuai dengan *real robot* yang diujikan, spesifikasi sensor dan aktuator yang digunakan oleh model robot di simulasi akan disesuaikan dengan spesifikasi komponen yang ada di prototipe robot ini. Seperti yang terlihat pada gambar 3.2, prototipe robot ini saat ini baru terdiri atas komponen *depth camera* dan *navigation*. Untuk komponen kamera, sebagai alternatif sementara, sebuah *webcam* dapat digunakan dan dipasang ke komputer robot melalui sambungan USB. Sedangkan untuk komponen *manipulator* untuk saat ini belum bisa digunakan.

Komponen *depth camera* yang digunakan di prototipe robot ini adalah Kinect V2 yang mampu menangkap citra berwarna dan citra kedalaman. Spesifikasi kedua citra yang ditangkap tersebut memiliki perbedaan pada sisi resolusi maksimal dan *field of view* (FoV), dimana citra berwarna bisa ditangkap hingga resolusi 1920 x 1080 dan FoV $84.1^\circ \times 53.8^\circ$, sedangkan citra kedalaman hanya bisa ditangkap pada resolusi 512 x 424 dan FoV $70.6^\circ \times 60^\circ$.

Komponen navigasi yang digunakan di prototipe robot ini adalah sekumpulan komponen elektronik yang digunakan di robot *IRIS* [18]. Komponen tersebut terdiri atas beberapa bagian seperti motor DC, *motor driver*, IMU, dsb. yang terhubung pada sebuah *controller* berbasis STM32F4. Nantinya, *controller* tersebut akan terhubung dengan komputer utama dari robot yang berbasis Intel NUC melalui sambungan *ethernet* dan komunikasi UDP (*user datagram protocol*).

Seperti yang dijelaskan sebelumnya, kamera yang digunakan di robot adalah sebuah *webcam* yang dipasang ke komputer robot melalui sambungan USB. Kamera tersebut saat ini belum ditetapkan untuk digunakan di prototipe robot, namun untuk memenuhi keperluan pengujian, maka kamera tersebut dipilih untuk digunakan. Kamera yang digunakan tersebut adalah Logitech C922 yang mampu digunakan untuk menangkap citra dengan resolusi hingga 1920 x 1080. Kamera tersebut dipilih karena kamera tersebut adalah kamera yang sama yang digunakan di robot *ICHIRO* [20] yang merupakan setengah bagian asal dari robot *Dienen* ini.

3.1.3 Struktur SDFFormat Robot

Dari paparan desain yang dijelaskan di bagian 3.1.1 dan paparan spesifikasi dari prototipe robot yang dijelaskan di bagian 3.1.2, model robot untuk simulasi kemudian dibuat menggunakan model 3D dari desain yang ada dan disusun berdasarkan bagian dan komponen yang ada pada desain tersebut.

Penyusunan SDFFormat dilakukan dengan memisahkan bagian yang dapat bergerak secara bebas di robot menjadi *link element* secara terpisah. Untuk setiap *link element* yang ada, beberapa *element* lain perlu ditambahkan ke *element* tersebut sehingga model

yang dibuat bisa digunakan di simulasi. Seperti yang terlihat pada potongan kode 3.1, setiap *link element* perlu memiliki *pose element* untuk menentukan posisi dari bagian robot yang diukur dari titik pusat robot, *inertial element* untuk menentukan berat dari bagian robot, *collision element* untuk menentukan *bounding box* yang digunakan oleh bagian robot, dan *visual element* yang digunakan untuk menentukan tampilan dari bagian robot tersebut.

Kode 3.1: *Link element* untuk bagian kepala robot.

```

1 <link name="head_link">
2   <pose>-0.012258 0 1.46717 0 0 0</pose>
3   <inertial>
4     <mass>1.0</mass>
5   </inertial>
6   <collision name="head_collision">
7     <geometry>
8       <mesh>
9         <uri>model://dienen_robot/collisions/head_collision.dae</uri>
10      </mesh>
11    </geometry>
12  </collision>
13  <visual name="head_visual">
14    <geometry>
15      <mesh>
16        <uri>model://dienen_robot/meshes/head.dae</uri>
17      </mesh>
18    </geometry>
19  </visual>
20 </link>

```

Agar terhubung satu sama lain, sebuah *joint element* perlu ditambahkan untuk menentukan hubungan antara dua *element*. Seperti yang terlihat pada potongan kode 3.2 dan 3.3, setiap *joint element* memiliki *type attribute* yang bisa bernilai *fixed* maupun *revolute*. *Joint element* dengan *type attribute* bernilai *fixed* digunakan untuk bagian yang tidak bisa bergerak bebas namun secara model terpisah seperti bagian bawah robot dan bagian atas robot, sedangkan *joint element* dengan *type attribute* bernilai *revolute* digunakan untuk bagian yang bisa bergerak berputar pada satu sumbu, seperti bagian *joint* yang ada di tangan.

Kode 3.2: *Joint element* yang menghubungkan bagian pundak kiri dan bagian badan atas robot.

```
1 <joint name="left_shoulder_joint" type="revolute">
2   <parent>upper_body_link</parent>
3   <child>left_shoulder_link</child>
4   <axis>
5     <xyz>0 -1 0</xyz>
6   </axis>
7 </joint>
```

Kode 3.3: *Joint element* yang menghubungkan bagian badan atas dan bagian badan bawah robot.

```
1 <joint name="upper_body_joint" type="fixed">
2   <parent>lower_body_link</parent>
3   <child>upper_body_link</child>
4 </joint>
```

Kode 3.4: *Sensor element* dari sensor kamera.

```
1 <link name="camera_link">
2   ...
3   <sensor name="camera" type="camera">
4     <pose>0.01 0 0 0 0 0</pose>
5     <update_rate>30</update_rate>
6     <camera>
7       <horizontal_fov>1.047198</horizontal_fov>
8       <image>
9         <width>1920</width>
10        <height>1080</height>
11      </image>
12      <clip>
13        <near>0.1</near>
14        <far>100</far>
15      </clip>
16    </camera>
17  </sensor>
18 </link>
```

Terakhir, penyusunan SDF format dilakukan dengan menambahkan sensor kamera dan *depth camera* yang digunakan pada robot. Sensor tersebut ditambahkan dengan menyematkan *sensor element* pada *link element* dari masing-masing bagian kamera dan *depth camera*. Seperti yang terlihat pada potongan kode 3.4 dan 3.5,

sensor kamera dan *depth camera* memiliki jumlah dan jenis *child element* yang sama, hanya berbeda di beberapa nilai dari setiap *element* dan nilai *type attribute* yang digunakan. Pada kamera, *sensor element* yang digunakan memiliki *type attribute* bernilai *camera*, sedangkan pada *depth camera*, *sensor element* yang digunakan memiliki *type attribute* bernilai *depth*. Sisanya, nilai setiap *element* yang ada di kedua sensor tersebut didapatkan dari spesifikasi kamera dan *depth camera* yang digunakan di prototipe robot seperti yang dijelaskan di bagian 3.1.2.

Kode 3.5: *Sensor element* dari sensor *depth camera*.

```

1 <link name="depth_camera_link">
2   ...
3   <sensor name="depth_camera" type="depth">
4     <pose>0.01 0 0 0 0 0</pose>
5     <update_rate>30</update_rate>
6     <camera>
7       <horizontal_fov>1.047198</horizontal_fov>
8       <image>
9         <width>512</width>
10        <height>424</height>
11      </image>
12      <clip>
13        <near>0.5</near>
14        <far>4.5</far>
15      </clip>
16    </camera>
17  </sensor>
18 </link>

```

3.2 Pengembangan Model Pengguna

Model pengguna yang ada di simulasi akan dibuat menyerupai bentuk manusia yang dapat berinteraksi dengan robot. Pada kondisi pengujian menggunakan robot *real*, pengguna akan berinteraksi dengan robot melalui *smart assistive posture device* yang digunakan untuk mengirimkan data ke robot yang berupa koordinat posisi dan orientasi pengguna, masukan suara yang diberikan pengguna melalui *voice recognition*, dan kondisi postur kaki dari pengguna, apakah sedang berdiri atau duduk. Dengan adanya model pengguna ini, pengujian yang ada di simulasi dapat dilakukan terhadap

model pengguna dengan perilaku yang sudah diprogram di awal, serta terhadap *real user* yang berinteraksi dengan model robot di simulasi melalui data yang dikirim dari *smart assistive posture device*.

Pada simulator Gazebo terdapat objek *Gazebo Actors* yang dapat digunakan untuk menampilkan model manusia yang dapat bergerak dan melakukan animasi seperti berjalan, duduk, dan lain sebagainya. Namun, objek tersebut sulit untuk diubah animasinya ketika simulasi sudah berjalan, sehingga kondisi perubahan postur kaki dari pengguna tidak bisa ditampilkan di model yang ada di simulasi. Oleh karena itu, pada penelitian ini model pengguna dibuat dengan cara yang sama dengan model robot dengan memisahkan setiap bagian pengguna yang dapat bergerak bebas menjadi *link element* dan kemudian akan saling terhubung menggunakan *joint element*.

3.2.1 Struktur SDFormat Pengguna

Kode 3.6: Struktur SDFormat dari model pengguna.

```
1 <?xml version="1.0" ?>
2 <sdf version="1.7">
3   <model name="dienen_human">
4
5     <link name="torso_link">
6       ...
7     </link>
8
9     <link name="left_upper_leg_link">
10      ...
11    </link>
12
13    ...
14
15    <joint name="left_upper_leg_joint" type="revolute">
16      ...
17    </joint>
18
19    ...
20
21  </model>
22 </sdf>
```

Struktur SDFFormat pada model pengguna dibuat dengan cara yang sama dengan yang dilakukan pada model robot. Seperti yang terlihat pada potongan kode 3.6, model pengguna akan terdiri dari beberapa *link element* yang berisi setiap bagian yang dapat bergerak bebas pada model pengguna seperti lengan atas, paha, leher, dan lain sebagainya. Kemudian, *link element* tersebut akan terhubung satu sama lain menggunakan *joint element*. Setiap *link element* maupun *joint element* yang ada nantinya juga memiliki *child element* lain seperti *collision element*, *axis element*, dsb. Sesuai dengan yang dibutuhkan oleh *link element* dan *joint element*.

3.3 Pengembangan Lingkungan Simulasi

Pengembangan lingkungan simulasi dilakukan dengan menyusun setiap model yang telah dibuat pada *file* SDFFormat. Terdapat dua jenis lingkungan simulasi yang akan digunakan pada penelitian ini, yakni lingkungan simulasi *outdoor* yang hanya berisi robot tanpa adanya objek lain, dan lingkungan simulasi *indoor* yang membentuk suatu ruangan tertutup dan berisi robot serta berbagai macam objek lain. Keduanya dibuat sebagai *Gazebo World* yang dapat berisi beberapa *child element* lain dan beberapa *include file* dari model yang sudah ada.

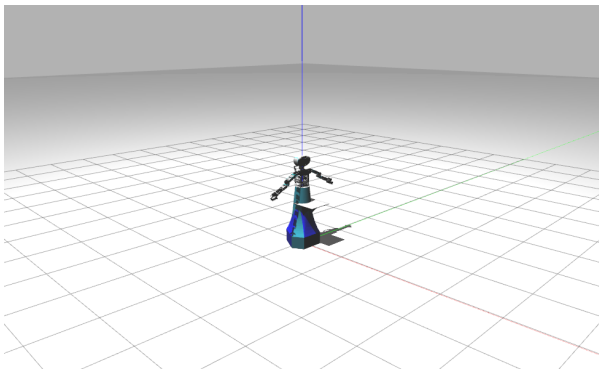
3.3.1 Lingkungan Simulasi *Outdoor*

Lingkungan simulasi *outdoor* dibuat dengan menyusun *file* SDFFormat yang berisi lingkungan paling dasar yang dibutuhkan robot agar pengujian masih bisa dilakukan di simulasi. Lingkungan simulasi ini dibuat untuk memudahkan pengujian gerakan pada robot sehingga robot dapat bergerak secara bebas tanpa terhalang oleh *obstacle* lain. Seperti yang terlihat pada potongan kode 3.7, lingkungan ini terdiri atas sebuah objek pencahayaan yang berupa *light element*, dan dua *include file* pada *path model://ground_plane* dan pada *path model://dienen_robot*. Kedua *include file* tersebut digunakan untuk memasukkan model *ground plane* yang sudah disediakan oleh Gazebo dan model robot yang telah dibuat sebelumnya di bagian 3.1. Di lingkungan ini, model *ground plane* diperlukan untuk memberikan *collision* pada dasar lingkungan sehingga model robot tidak jatuh terus menerus ke bawah karena pengaruh gravitasi di simulasi.

Kode 3.7: Struktur SDFFormat dari lingkungan simulasi *outdoor*.

```
1 <?xml version="1.0"?>
2 <sdf version="1.7">
3   <world name="outdoor">
4
5     <light name="sun" type="directional">
6       ...
7     </light>
8
9     <include>
10      <uri>model://ground_plane</uri>
11    </include>
12
13    <include>
14      <pose>0.0 0 1.0 0 0 0</pose>
15      <uri>model://dienen_robot</uri>
16    </include>
17
18  </world>
19 </sdf>
```

Setelah *file* SDFFormat dibuat, hasil yang didapatkan setelah dijalankan pada simulator tampak seperti pada gambar 3.3. Sesuai dengan susunan yang diisi pada *file* SDFFormat, tampak adanya model robot dengan objek *ground plane* yang ada di bawah, sedangkan objek pencahayaan tampak dengan adanya bayangan dari robot dan gradasi warna cerah pada objek *ground plane*.



Gambar 3.3: Tampilan lingkungan simulasi *outdoor* pada Gazebo.

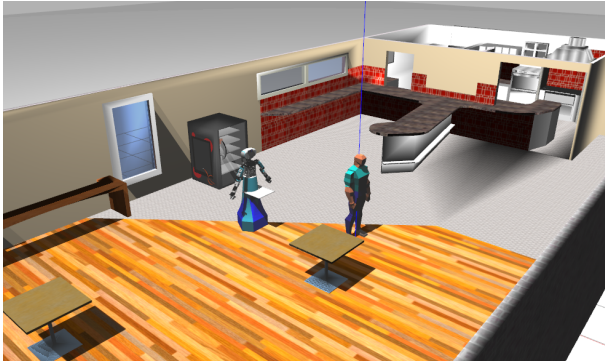
3.3.2 Lingkungan Simulasi *Indoor*

Lingkungan simulasi *indoor* dibuat dengan menyusun *file* SDF format yang berisi ruangan tertutup dengan berbagai macam objek di dalamnya. Seperti yang terlihat pada potongan kode 3.8, lingkungan ini terdiri atas objek pencahayaan dan *include file* pada *path* `model://cafe`, `model://cafe_table`, `model://dienen_robot`, dan `model://dienen_human`. Dua *path* pertama merupakan *path* dari model yang sudah disediakan oleh Gazebo, yang membentuk lingkungan kafe beserta perabotan yang ada di dalamnya, sedangkan dua *path* terakhir merupakan *path* dari model robot dan model pengguna yang sebelumnya sudah dibuat di bagian 3.1 dan 3.2.

Kode 3.8: Struktur SDF format dari lingkungan simulasi *indoor*.

```
1 <?xml version="1.0"?>
2 <sdf version="1.7">
3   <world name="indoor">
4
5     <light name="sun" type="directional">
6       ...
7     </light>
8
9     <include>
10      <uri>model://cafe</uri>
11    </include>
12
13    <include>
14      <pose>0.5 -1.6 0.2 0 0 0</pose>
15      <uri>model://cafe_table</uri>
16    </include>
17
18    ...
19
20    <include>
21      <pose>-2 -1 1.0 0 0 0</pose>
22      <uri>model://dienen_robot</uri>
23    </include>
24
25    <include>
26      <pose>0 0 1.0 0 0 0</pose>
27      <uri>model://dienen_human</uri>
28    </include>
29
30   </world>
31 </sdf>
```

Setelah *file* SDFormat dibuat, hasil yang didapatkan setelah dijalankan pada simulator tampak seperti pada gambar 3.4. Sesuai dengan susunan yang diisi pada *file* SDFormat, tampak adanya model robot dan pengguna di dalam sebuah ruangan kafe dengan berbagai macam perabotan yang ada di dalamnya seperti meja, brankas, dan lain sebagainya.

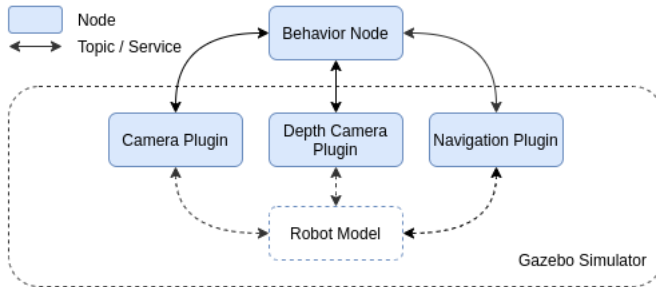


Gambar 3.4: Tampilan lingkungan simulasi *indoor* pada Gazebo.

3.4 Integrasi *Plugin* Untuk Simulasi

Agar sistem yang ada di simulasi dapat bekerja dan terabstraksi selayaknya sistem yang ada pada *real robot*, model yang ada di simulasi perlu diintegrasikan dengan *plugin* yang memungkinkan sensor dan aktuator yang ada di simulasi berkomunikasi dengan *node* lain menggunakan sistem komunikasi antar-proses ROS 2. Agar dapat bekerja dengan model yang telah dibuat, *plugin* tersebut nantinya perlu disematkan pada *file* SDFormat dari model robot sebagai *child element* dalam bentuk *plugin element*.

Seperti yang terlihat pada gambar 3.5, model robot akan diintegrasikan dengan tiga buah *plugin*, *camera plugin* yang akan terabstraksi sebagai komponen kamera, *depth camera plugin* yang akan terabstraksi sebagai komponen *depth camera*, dan *navigation plugin* yang akan terabstraksi sebagai komponen navigasi. Ketiga *plugin* tersebut nantinya akan dibentuk sebagai *ROS 2 node* sehingga me-



Gambar 3.5: Diagram integrasi *plugin* untuk simulasi.

mungkinkan komunikasi dengan *node* lain, termasuk *behavior node* yang akan mengatur tingkah laku dari robot selama pengujian.

3.4.1 *Navigation Plugin*

Kode 3.9: *Class* dari *navigation plugin*.

```

1 #include <gazebo/common/Plugin.hh>
2 #include <rclcpp/rclcpp.hpp>
3 ...
4
5 namespace dienen_gazebo_plugins
6 {
7
8 class NavigationPlugin : public gazebo::ModelPlugin
9 {
10 public:
11     ...
12
13 private:
14     rclcpp::Node::SharedPtr node;
15
16     rclcpp::Publisher<Odometry>::SharedPtr odometry_pub;
17     rclcpp::Subscription<Twist>::SharedPtr twist_sub;
18
19     ...
20 };
21
22 } // namespace dienen_gazebo_plugins

```

Navigation plugin merupakan *Gazebo plugin* yang digunakan untuk mengabstraksi komponen navigasi di simulasi. *Plugin* ini ditulis dalam bahasa C++ dan dibuat menjadi *class* dengan nama `dienen_gazebo_plugins::NavigationPlugin`. Seperti yang terlihat pada potongan kode 3.9, *class* ini dibuat dengan menurunkan *class* `gazebo::ModelPlugin` sebagai *parent class* dari *class* ini sehingga memungkinkan *class* ini untuk mengakses maupun memanipulasi data yang ada pada *Gazebo model*.

Plugin ini digunakan untuk mengirimkan estimasi posisi dan orientasi dari robot sebagai odometri melalui *topic /odom* dan menerima masukan kecepatan (*twist*) melalui *topic /cmd_vel*. Odometri yang dikirimkan didapatkan dari posisi dan orientasi mutlak dari model robot yang ada di simulasi, sedangkan masukan kecepatan yang diterima akan diterjemahkan sebagai memberikan *force* kepada model robot yang mengakibatkan model tersebut bergerak ke arah yang sesuai dengan *force* yang diberikan.

Setelah *plugin* dibuat, file SDFFormat dari model robot perlu diubah dengan menyematkan *plugin element* di file tersebut. Seperti yang terlihat pada potongan kode 3.10, sebuah *plugin element* ditambahkan di model robot dengan *name attribute* berupa nama *plugin* sekaligus nama *node* yang akan digunakan dan *filename attribute* yang merujuk pada nama *shared library* dari *plugin* yang telah dikompilasi.

Kode 3.10: Integrasi *navigation plugin* pada model robot.

```
1 <?xml version="1.0" ?>
2 <sdf version="1.7">
3   <model name="dienen_robot">
4
5     <plugin name="navigation_plugin" filename="↵
6       libdienen_navigation_plugin.so" />
7
8     ...
9   </model>
10 </sdf>
```

3.4.2 Camera Plugin

Camera plugin merupakan *Gazebo plugin* yang digunakan untuk mengabstraksi komponen kamera di simulasi. *Plugin* yang digunakan ini akan mengirimkan data citra melalui *topic /image_raw* dan mengirimkan informasi kamera seperti resolusi citra, FoV, matriks proyeksi, dan lain sebagainya melalui *topic /camera_info*. *Plugin* ini didapatkan dari *package gazebo_ros*, sehingga yang perlu dilakukan hanyalah mengintegrasikan *plugin* tersebut ke model robot yang sudah dibuat di bagian 3.1.

Kode 3.11: Integrasi *camera plugin* pada model robot.

```
1 <link name="camera_link">
2   ...
3   <sensor name="camera" type="camera">
4     ...
5     <plugin name="camera_plugin" filename="↵
      libgazebo_ros_camera.so">
6       <camera_name>camera</camera_name>
7     </plugin>
8   </sensor>
9 </link>
```

Seperti yang terlihat pada potongan kode 3.11, sebuah *plugin element* ditambahkan sebagai *child element* dari *sensor element*. Sama seperti *plugin* sebelumnya, *plugin* ini juga menggunakan *name attribute* untuk menentukan nama node dan *filename attribute* untuk menentukan *shared library* dari *plugin* yang telah dikompilasi. Dalam hal ini *plugin* yang digunakan adalah *plugin gazebo_ros_camera* dari *package gazebo_ros*. Sebagai tambahan, *plugin* ini menggunakan *camera name element* untuk menentukan *namespace* dari *topic* yang digunakan, sehingga dengan *camera name element* bernilai *camera* maka *topic* yang digunakan adalah */camera*↵
/image_raw dan */camera/camera_info*.

3.4.3 Depth Camera Plugin

Depth camera plugin merupakan *Gazebo plugin* yang digunakan untuk mengabstraksi komponen *depth camera* di simulasi. Sama seperti yang digunakan pada *camera plugin*, *plugin* ini juga menggunakan *plugin gazebo_ros_camera* dari *package gazebo_ros*. Hanya saja

plugin ini mengirimkan dua macam data citra, yakni citra berwarna dengan data citra yang dikirim melalui *topic /image_raw* dan informasi kamera yang dikirim melalui *topic /camera_info*, serta citra kedalaman (*depth image*) dengan data citra yang dikirim melalui *topic /depth/image_raw* dan informasi kamera yang dikirim melalui *topic /depth/camera_info*.

Kode 3.12: Integrasi *depth camera plugin* pada model robot.

```

1 <link name="depth_camera_link">
2   ...
3   <sensor name="depth_camera" type="depth">
4     ...
5     <plugin name="depth_camera_plugin" filename="↵
        libgazebo_ros_camera.so">
6       <camera_name>depth_camera</camera_name>
7     </plugin>
8   </sensor>
9 </link>

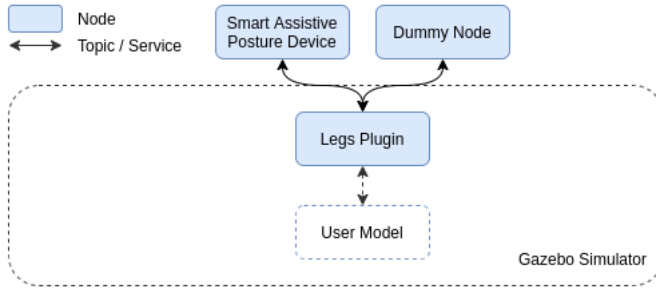
```

Seperti yang terlihat pada potongan kode 3.12, cara yang sama seperti yang ada pada *camera plugin* di bagian 3.4.2 juga dilakukan pada *plugin* ini. Perbedaannya terletak pada *name attribute* yang digunakan dan nilai dari *camera name* yang menentukan *namespace* dari *topic* yang digunakan.

3.4.4 Legs Plugin

Legs plugin merupakan *Gazebo plugin* yang digunakan untuk menghubungkan model pengguna dengan sistem komunikasi antarproses ROS 2. Seperti yang terlihat pada gambar 3.6, *plugin* ini dibuat agar model pengguna yang ada di simulasi bisa terabstraksi untuk diakses dan dimanipulasi oleh *smart assistive posture device* maupun *dummy node* yang mengirimkan data yang sama dengan yang dikirim oleh *smart assistive posture device*.

Legs plugin memiliki dua kegunaan utama. Yang pertama adalah untuk mengubah posisi dan orientasi dari model pengguna sesuai dengan posisi dan orientasi yang diterima dari perhitungan *smart assistive posture device* maupun *dummy node*. Sedangkan yang kedua adalah untuk mengubah posisi *joints* di kaki menjadi duduk jongkok maupun berdiri sesuai dengan nilai postur kaki yang dikirim oleh *smart assistive posture device* maupun *dummy node*.



Gambar 3.6: Diagram integrasi *plugin* untuk model pengguna di simulasi.

Kode 3.13: *Class* dari *legs plugin*.

```

1 #include <beine_cpp/beine_cpp.hpp>
2 #include <gazebo/common/Plugin.hh>
3 #include <rclcpp/rclcpp.hpp>
4
5 ...
6
7 namespace beine_gazebo_plugins
8 {
9
10 class LegsPlugin : public gazebo::ModelPlugin
11 {
12 public:
13     LegsPlugin();
14
15     ...
16
17 private:
18     rclcpp::Node::SharedPtr node;
19
20     std::shared_ptr<beine_cpp::LegsConsumer> ↵
        legs_consumer;
21
22     ...
23 };
24
25 } // namespace beine_gazebo_plugins

```

Sama seperti *navigation plugin* yang ada di bagian 3.4.1, seperti yang terlihat pada potongan kode 3.13, *plugin* ini juga ditulis dalam bahasa C++ dan dibuat dengan menurunkan *class gazebo::ModelPlugin* sebagai *parent class* dari *class* ini. *Plugin* ini menggunakan objek *beine_cpp::LegsConsumer* yang memudahkan *subscription* dari *topic* yang berhubungan dengan data yang dikirim oleh *smart assistive posture device*. Data tersebut berupa data posisi yang dikirim melalui *topic /position*, data orientasi yang dikirim melalui *topic /orientation*, data perintah suara yang dikirim melalui *topic /command*, dan data postur kaki yang dikirim melalui *topic /stance*.

Setelah *plugin* dibuat, file SDF format dari model robot perlu diubah dengan menyematkan *plugin element* di file tersebut. Seperti yang terlihat pada potongan kode 3.14, *plugin element* disematkan sebagai *child element* dari *model element*. Pada *legs plugin*, beberapa *child element* lain perlu disematkan pada *plugin* tersebut, seperti *joint force strength element* dan *joint force smoothness element* yang menentukan bagaimana transisi postur kaki terjadi, serta *left hip pitch joint element*, *left knee pitch joint element*, dan lain sebagainya yang menentukan *joint element* yang akan diubah ketika terjadi transisi pada postur kaki pengguna.

Kode 3.14: Integrasi *legs plugin* pada model pengguna.

```

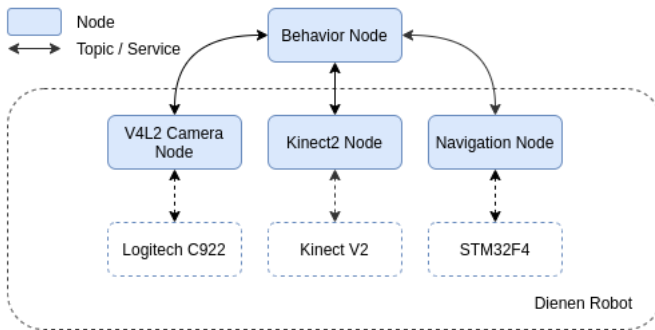
1 <?xml version="1.0" ?>
2 <sdf version="1.7">
3   <model name="dienen_human">
4     <plugin name="dienen_human_legs" filename="↵
      libbeine_legs_plugin.so">
5       <joint_force_strength>1000.0</↵
        joint_force_strength>
6       <joint_force_smoothness>0.2</↵
        joint_force_smoothness>
7       <left_hip_pitch_joint>left_upper_leg_joint</↵
        left_hip_pitch_joint>
8       <left_knee_pitch_joint>left_lower_leg_joint</↵
        left_knee_pitch_joint>
9       ...
10    </plugin>
11
12    ...
13  </model>
14 </sdf>

```

3.5 Pengembangan *Behavior Node*

Behavior node merupakan *ROS 2 node* yang dibuat untuk mengatur tindakan yang akan dilakukan oleh SARs berdasarkan data yang diterima dari *node* lain. Seperti yang dijelaskan di bagian 3.4, *behavior node* akan terhubung dengan *node* lain yang secara abstrak merepresentasikan komponen yang dimiliki oleh robot *Dienen* seperti penjelasan di bagian 3.1. *Behavior node* dijalankan secara terpisah dari *node* yang mengatur komponen yang ada di robot seperti komponen kamera, *depth camera*, *manipulator*, dan navigasi. Terpisahnya *behavior node* ini dilakukan untuk membuat pengujian pada SARs bisa dilakukan menggunakan berbagai macam data yang diterima, baik itu data *real-time* yang berasal dari *real robot*, data yang direkam dari robot, maupun data yang ada di simulasi.

3.6 Integrasi Sistem pada *Real Robot*



Gambar 3.7: Diagram integrasi sistem pada *real robot*.

Integrasi sistem pada *real robot* dapat dilakukan dengan mengganti *node* yang digunakan di simulasi dengan *node* yang mengakses komponen yang ada pada *real robot*. Seperti yang terlihat pada gambar 3.7, *behavior node* yang digunakan masih sama, perbedaannya adalah digantinya *camera plugin*, *depth camera plugin*, dan *navigation plugin* dengan *V4L2 camera node*, *Kinect2 node*, dan

navigation node. Penggantian ini dapat dilakukan dengan mudah, karena seperti yang dijelaskan di bagian 3.5, *behavior node* mengakses setiap *node* yang mewakili komponen pada robot secara abstrak, yang mana keduanya dianggap *node* yang sama oleh *behavior node* terlepas dari bagaimana dan darimana data tersebut berasal, baik dari simulasi maupun dari *real robot*.

V4L2 camera node merupakan *ROS 2 node* yang digunakan untuk mengakses kamera yang ada di sistem operasi *Linux* dan mengirimkan data citra beserta informasi kamera melalui *ROS 2 topic*. *Kinect2 node* merupakan *ROS 2 node* yang digunakan untuk mengakses *Kinect V2* menggunakan *libfreenect2* dan mengirimkan data citra berwarna, citra kedalaman (*depth image*), dan informasi kamera melalui *ROS 2 topic*. Sedangkan *navigation node* adalah *ROS 2 node* yang digunakan untuk mengakses *STM32F4 controller* yang telah diprogram dengan sistem navigasi yang ada di robot *Dienen*. *Navigation node* mengakses data yang ada di *controller* tersebut menggunakan komunikasi UDP dan kemudian disalurkan oleh *navigation node* melalui *ROS 2 topic* sehingga bisa diakses oleh *node* lain.

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan dipaparkan hasil pengujian serta analisis dari desain dan implementasi sistem yang telah dibuat sebelumnya di bab 3. Setiap pengujian yang dilakukan pada penelitian ini diujikan menggunakan model robot yang ada di simulasi dan menggunakan prototipe robot yang diuji secara *real*.

Pengujian yang dilakukan menggunakan model robot yang ada di simulasi dilakukan di lingkungan *outdoor* dan di lingkungan *in-door* seperti yang telah dibuat di bagian 3.3. Pengujian tersebut dilakukan di simulator Gazebo dengan menggunakan komputer dengan spesifikasi seperti yang dapat dilihat pada tabel 4.1. Sedangkan pengujian yang dilakukan menggunakan prototipe robot secara *real* dilakukan di lingkungan laboratorium AJ403 Teknik Komputer ITS serta menggunakan komputer robot dengan spesifikasi seperti yang dapat dilihat pada tabel 4.2.

Tabel 4.1: Spesifikasi komputer untuk menjalankan simulator.

| | |
|-----|-----------------------------|
| OS | Ubuntu 20.04.2 LTS |
| CPU | Intel i3-8100 (4) @ 3.600GH |
| GPU | NVIDIA GeForce GTX 1050 Ti |
| RAM | 7901 MiB |

Tabel 4.2: Spesifikasi komputer yang ada pada prototipe robot.

| | |
|-----|-----------------------------|
| OS | Ubuntu 20.04.2 LTS |
| CPU | Intel i3-10110U (4) @ 4.100 |
| GPU | Intel UHD Graphics |
| RAM | 3648 MiB |

4.1 Pengujian Gerakan

Pengujian gerakan terbagi menjadi dua bagian, yakni pengujian gerakan linier dan pengujian gerakan putar (*angular movement*). Pengujian gerakan ini dilakukan untuk menguji perintah gerakan yang dikirim melalui *topic /cmd_vel* serta untuk menguji estimasi posisi dan orientasi dari perhitungan odometri yang dapat diterima melalui *topic /odom*. Kedua pengujian ini melibatkan komponen navigasi yang ada pada robot dan *move for node* sebagai *behavior node* yang mengatur tingkah laku robot. Dalam hal ini, tingkah laku yang dilakukan adalah memberikan perintah gerakan pada robot selama selang waktu tertentu.

4.1.1 Pengujian Gerakan Linier dan Estimasi Posisi di Simulasi



Gambar 4.1: Relasi antar-*node* dari pengujian gerakan linier dan estimasi posisi di simulasi.

Pengujian gerakan linear dan estimasi posisi di simulasi dilakukan dengan cara menjalankan lingkungan *outdoor* pada simulator Gazebo, menjalankan *move for node* sebagai *behavior node* dari pengujian, dan menjalankan *odometry echo node* untuk menampilkan hasil estimasi posisi dari kalkulasi odometri robot. Seperti yang terlihat pada gambar 4.1, *node /move_for* akan mengirimkan *topic /cmd_vel* yang memerintahkan *node /navigation_plugin* untuk menggerakkan robot sesuai dengan data kecepatan yang ada di *topic* tersebut, setelah itu *node /odometry_echo* akan menerima *topic /odom* yang menunjukkan estimasi posisi dari perhitungan yang dilakukan oleh *node /navigation_plugin*.

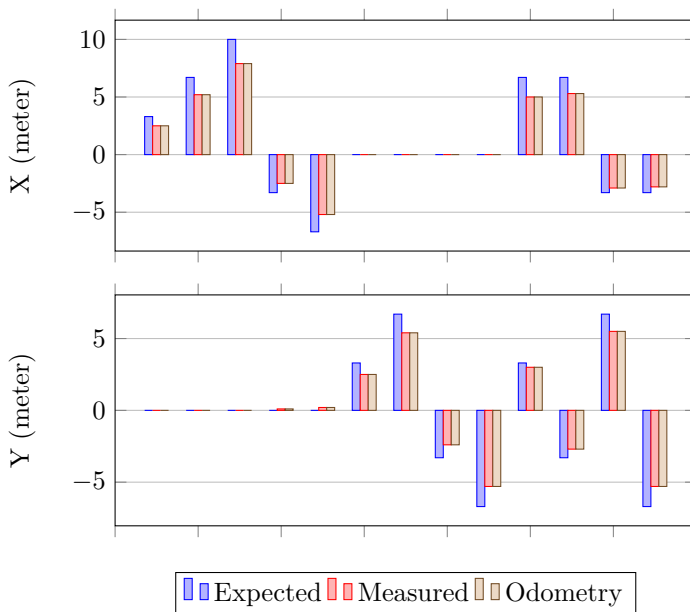
Pengujian ini dilakukan dengan berbagai macam konfigurasi kecepatan X dan Y yang diperintahkan selama 10 detik. Hasil pengujian ini bisa dilihat pada tabel 4.3. Pada tabel tersebut, nilai yang ada di kolom *speed* adalah besar kecepatan yang diatur pada

Tabel 4.3: Hasil estimasi posisi dari gerakan linier di simulasi selama 10 detik.

| Speed | | Expected Position | | Measured Position | | Odometry Position | |
|-----------|-----------|-------------------|-------|-------------------|-------|-------------------|-------|
| x (m/min) | y (m/min) | x (m) | y (m) | x (m) | y (m) | x (m) | y (m) |
| 20 | 0 | 3.3 | 0.0 | 2.5 | 0.0 | 2.5 | 0 |
| 40 | 0 | 6.7 | 0.0 | 5.2 | 0.0 | 5.2 | 0 |
| 60 | 0 | 10.0 | 0.0 | 7.9 | 0.0 | 7.9 | 0 |
| -20 | 0 | -3.3 | 0.0 | -2.5 | 0.1 | -2.5 | 0.1 |
| -40 | 0 | -6.7 | 0.0 | -5.2 | 0.2 | -5.2 | 0.2 |
| 0 | 20 | 0.0 | 3.3 | 0.0 | 2.5 | 0 | 2.5 |
| 0 | 40 | 0.0 | 6.7 | 0.0 | 5.4 | 0 | 5.4 |
| 0 | -20 | 0.0 | -3.3 | 0.0 | -2.4 | 0 | -2.4 |
| 0 | -40 | 0.0 | -6.7 | 0.0 | -5.3 | 0 | -5.3 |
| 40 | 20 | 6.7 | 3.3 | 5.0 | 3.0 | 5 | 3 |
| 40 | -20 | 6.7 | -3.3 | 5.3 | -2.7 | 5.3 | -2.7 |
| -20 | 40 | -3.3 | 6.7 | -2.9 | 5.5 | -2.9 | 5.5 |
| -20 | -40 | -3.3 | -6.7 | -2.8 | -5.3 | -2.8 | -5.3 |

topic /cmd_vel, nilai yang ada di kolom *estimated position* didapatkan dari perkalian besar kecepatan dengan durasi pengujian, nilai yang ada di kolom *measured position* didapatkan dari koordinat model yang ada di simulasi, dan terakhir nilai yang ada di kolom *odometry position* didapatkan dari data yang ada pada *topic /odom*.

Dari data yang dihasilkan oleh pengujian ini dapat diketahui bahwa gerakan yang diperintahkan kepada robot cenderung menghasilkan posisi robot yang sesuai dengan posisi perkiraan (*estimated position*). Lebih lanjut, ketika hasil tersebut ditampilkan sebagai grafik, seperti yang terlihat pada gambar 4.2, hasil yang didapatkan cenderung memiliki arah positif negatif yang sesuai dengan yang diharapkan (*expected*) walaupun dengan *error* jarak yang relatif tidak kecil (kurang lebih 1-2 meter).



Gambar 4.2: Grafik estimasi posisi dari gerakan linier di simulasi.

4.1.2 Pengujian Gerakan Linier dan Estimasi Posisi pada *Real Robot*

Pengujian gerakan linier dan estimasi posisi pada *real robot* dilakukan dengan cara yang sama dengan yang dilakukan di simulasi seperti yang ada di bagian 4.1.1. Perbedaannya, pada pengujian ini komponen navigasi yang sebelumnya berasal dari simulasi digantikan dengan *navigation node* yang mengakses perangkat yang ada pada *real robot*. Seperti yang terlihat pada gambar 4.3, *node /move_for* akan mengirimkan *topic /cmd_vel* dengan fungsi yang sama ke *node /navigation*, setelah itu *node /odometry_echo* akan menerima *topic /odom* yang berasal dari *node /navigation*.



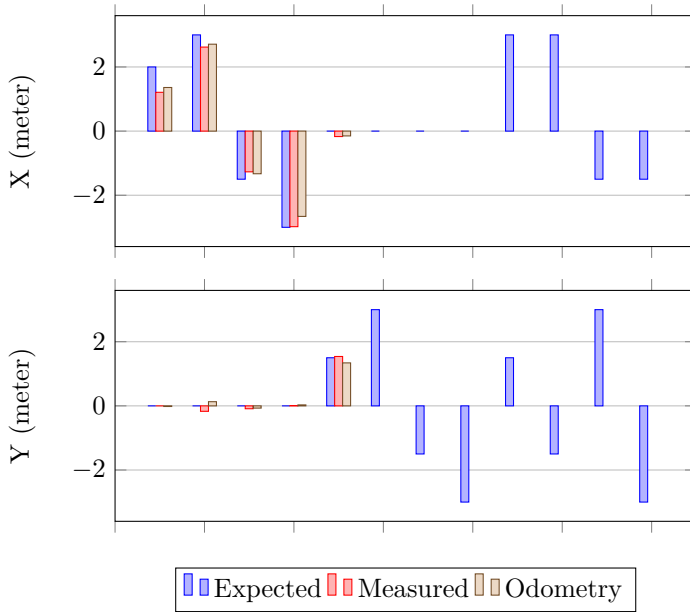
Gambar 4.3: Relasi antar-*node* dari pengujian gerakan linier dan estimasi posisi pada *real robot*.

Pengujian pada *real robot* ini juga dilakukan dengan berbagai macam konfigurasi kecepatan X dan Y yang diperintahkan selama 3 detik. Hasil pengujian ini bisa dilihat pada tabel 4.4. Pada tabel tersebut, nilai yang ada di kolom *speed* adalah besar kecepatan yang diatur pada *topic /cmd_vel*, nilai yang ada di kolom *estimated position* didapatkan dari perkalian besar kecepatan dengan durasi pengujian, nilai yang ada di kolom *measured position* didapatkan dari pengukuran perpindahan *real robot* menggunakan meter ukur, dan terakhir nilai yang ada di kolom *odometry position* didapatkan dari data yang ada di *topic /odom*.

Seperti kesimpulan pada pengujian sebelumnya, dari data yang dihasilkan oleh pengujian ini juga dapat diketahui bahwa gerakan yang diperintahkan robot cenderung menghasilkan posisi robot yang sesuai dengan posisi perkiraan (*estimated position*). Lebih lanjut, ketika hasil tersebut ditampilkan sebagai grafik, seperti yang terlihat pada gambar 4.4, hasil yang didapatkan juga cenderung memiliki arah positif dan negatif yang sesuai dengan yang diharapkan (*expected*) dan sama dengan yang dihasilkan pada pengujian sebelumnya.

Tabel 4.4: Hasil estimasi posisi dari gerakan linier pada *real robot* selama 3 detik.

| Speed | | Expected Position | | Measured Position | | Odometry Position | |
|---------|---------|-------------------|-------|-------------------|-------|-------------------|-------|
| x (m/s) | y (m/s) | x (m) | y (m) | x (m) | y (m) | x (m) | y (m) |
| 0.5 | 0.0 | 2 | 0 | 1.21 | 0 | 1.36 | -0.02 |
| 1.0 | 0.0 | 3 | 0 | 2.62 | -0.17 | 2.71 | 0.13 |
| -0.5 | 0.0 | -1.5 | 0 | -1.27 | -0.09 | -1.33 | -0.07 |
| -1.0 | 0.0 | -3 | 0 | -2.98 | 0.01 | -2.66 | 0.03 |
| 0.0 | 0.5 | 0 | 1.5 | -0.17 | 1.54 | -0.15 | 1.34 |
| 0.0 | 1.0 | 0 | 3 | | | | |
| 0.0 | -0.5 | 0 | -1.5 | | | | |
| 0.0 | -1.0 | 0 | -3 | | | | |
| 1.0 | 0.5 | 3 | 1.5 | | | | |
| 1.0 | -0.5 | 3 | -1.5 | | | | |
| -0.5 | 1.0 | -1.5 | 3 | | | | |
| -0.5 | -1.0 | -1.5 | -3 | | | | |



Gambar 4.4: Grafik estimasi posisi dari gerakan linier pada *real robot*.

4.1.3 Pengujian Gerakan Putar dan Estimasi Orientasi di Simulasi

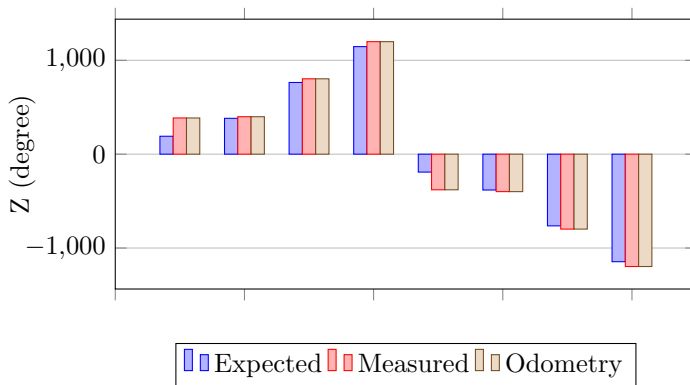
Pengujian gerakan putar dan estimasi orientasi di simulasi dilakukan dengan menjalankan *node* yang sama seperti yang ada pada pengujian di bagian 4.1.1. Perbedaannya, pengujian ini dilakukan pada beberapa nilai kecepatan putar di sumbu Z selama 10 detik.

Hasil pengujian ini bisa dilihat pada tabel 4.5. Pada tabel tersebut, nilai yang ada di kolom *speed* adalah besar kecepatan yang diatur pada *topic /cmd/vel*, nilai yang ada di kolom *estimated* didapatkan dari perkalian besar kecepatan putar dengan durasi pengujian, nilai yang ada di kolom *measured* didapatkan dari orientasi model yang ada di simulasi, dan terakhir nilai yang ada di kolom *odometry* didapatkan dari data orientasi yang ada pada *topic /odom*.

Tabel 4.5: Hasil estimasi orientasi dari gerakan putar di simulasi selama 10 detik.

| Speed | Expected | Measured | Odometry |
|-------------|----------|----------|----------|
| Z (rad/min) | Z (deg) | Z (deg) | Z (deg) |
| 20 | 191.0 | 385.7 | 385.7 |
| 40 | 382.0 | 399.0 | 399.0 |
| 80 | 763.9 | 802.9 | 802.9 |
| 120 | 1145.9 | 1198.7 | 1198.7 |
| -20 | -191.0 | -379.6 | -379.6 |
| -40 | -382.0 | -399.2 | -399.2 |
| -80 | -763.9 | -798.4 | -798.4 |
| -120 | -1145.9 | -1197.1 | -1197.1 |

Dari data yang dihasilkan oleh pengujian ini dapat diketahui bahwa gerakan putar yang diperintahkan kepada robot cenderung menghasilkan orientasi robot yang sesuai dengan orientasi perkiraan (*estimated orientation*). Lebih lanjut, ketika hasil tersebut ditampilkan sebagai grafik, seperti yang terlihat pada gambar 4.5, hasil yang didapatkan cenderung memiliki arah positif negatif yang sesuai dengan yang diharapkan (*expected*).



Gambar 4.5: Grafik estimasi orientasi dari gerakan putar di simulasi.

4.1.4 Pengujian Gerakan Putar dan Estimasi Orientasi pada *Real Robot*

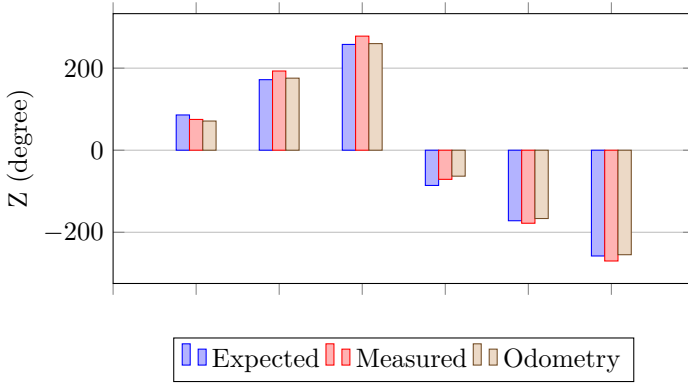
Sama seperti pada pengujian simulasi di bagian sebelumnya, Pengujian gerakan putar dan estimasi orientasi pada *real robot* juga dilakukan dengan menjalankan *node* yang sama seperti yang ada pada pengujian gerakan linier di bagian 4.1.2. Perbedaannya, pengujian ini dilakukan pada beberapa kecepatan putar di sumbu Z selama 3 detik.

Hasil pengujian ini bisa dilihat pada tabel 4.6. Pada tabel tersebut, nilai yang ada di kolom *speed* adalah besar kecepatan yang diatur pada *topic /cmd/vel*, nilai yang ada di kolom *estimated* didapatkan dari perkalian besar kecepatan putar dengan durasi pengujian, nilai yang ada di kolom *measured* didapatkan dari pengukuran menggunakan kompas yang dipasang pada robot, dan terakhir nilai yang ada di kolom *odometry* didapatkan dari data orientasi yang ada pada *topic /odom*.

Tabel 4.6: Hasil estimasi orientasi dari gerakan putar pada *real robot* selama 3 detik.

| Speed | Expected | Measured | Odometry |
|-----------|----------|----------|----------|
| Z (rad/s) | Z (deg) | Z (deg) | Z (deg) |
| 0.5 | 85.9 | 75.0 | 71.1 |
| 1.0 | 171.9 | 193.0 | 175.6 |
| 1.5 | 257.8 | 278.0 | 259.8 |
| -0.5 | -85.9 | -71.0 | -63.3 |
| -1.0 | -171.9 | -178.0 | -166.5 |
| -1.5 | -257.8 | -270.0 | -254.7 |

Seperti kesimpulan pada pengujian sebelumnya, dari data yang dihasilkan oleh pengujian ini juga dapat diketahui bahwa gerakan putar yang diperintahkan kepada robot cenderung menghasilkan orientasi robot yang sesuai dengan orientasi perkiraan (*estimated orientation*). Lebih lanjut, ketika hasil tersebut ditampilkan sebagai grafik, seperti yang terlihat pada gambar 4.6, hasil yang didapatkan juga cenderung memiliki arah positif negatif yang sesuai dengan yang diharapkan (*expected*).



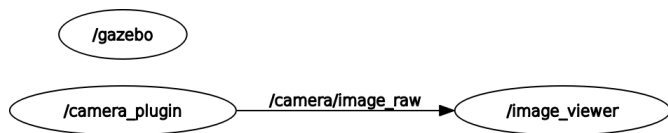
Gambar 4.6: Grafik estimasi orientasi dari gerakan putar pada *real robot*.

4.2 Pengujian Citra Kamera

Pengujian citra kamera terbagi menjadi dua bagian, yakni pengujian pengiriman citra kamera di perangkat yang sama dan pengujian pengiriman citra kamera antara perangkat yang berbeda. Pengujian pengiriman citra ini dilakukan untuk menguji data citra gambar yang diterima melalui *topic /raw_image* serta untuk menguji kemampuan ROS 2 dalam mengirim data citra dengan ukuran besar secara *real-time*. Kedua pengujian ini melibatkan komponen kamera yang ada pada robot, *image viewer node* untuk melihat tampilan data citra yang diterima, dan *command line \$ ros2 topic* yang digunakan untuk mengukur estimasi *delay* dan frekuensi dari data citra yang diterima.

4.2.1 Pengujian Pengiriman Citra Kamera di Simulasi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et ma-



Gambar 4.7: Relasi antar-*node* dari pengujian pengiriman citra kamera di simulasi.

lesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

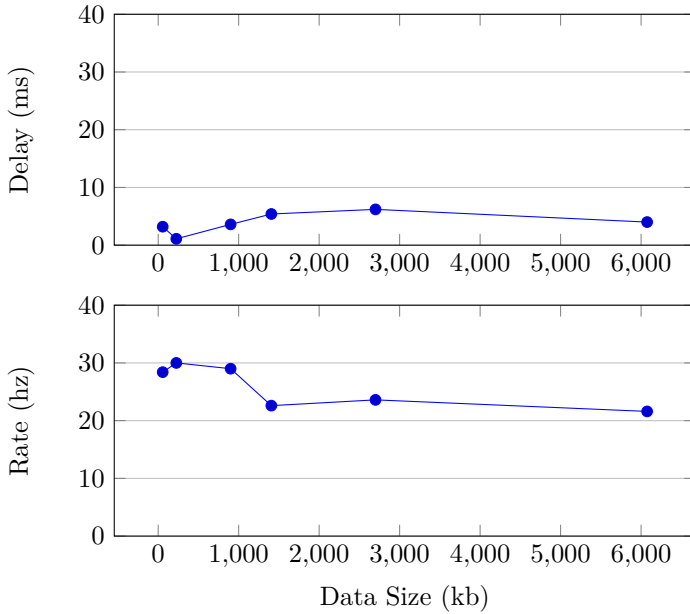
Tabel 4.7: Hasil *delay* dan frekuensi dari pengiriman citra kamera di simulasi.

| Width | Height | Size (kb) | Delay (ms) | Rate (hz) |
|-------|--------|-----------|------------|-----------|
| 160 | 120 | 56 | 3.2 | 28.4 |
| 320 | 240 | 225 | 1.1 | 30.0 |
| 640 | 480 | 900 | 3.6 | 29.0 |
| 800 | 600 | 1406 | 5.4 | 22.6 |
| 1280 | 720 | 2700 | 6.2 | 23.6 |
| 1920 | 1080 | 6075 | 4.0 | 21.6 |

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

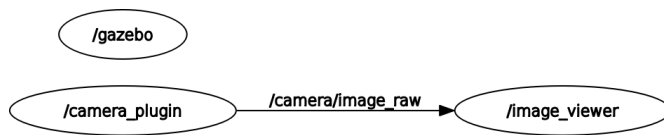
4.2.2 Pengujian Pengiriman Citra Kamera pada *Real Robot*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam orci libero nonummy erat.



Gambar 4.8: Grafik *delay* dan frekuensi dari pengiriman citra di simulasi.

que. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



Gambar 4.9: Relasi antar-*node* dari pengujian pengiriman citra kamera pada *real robot*.

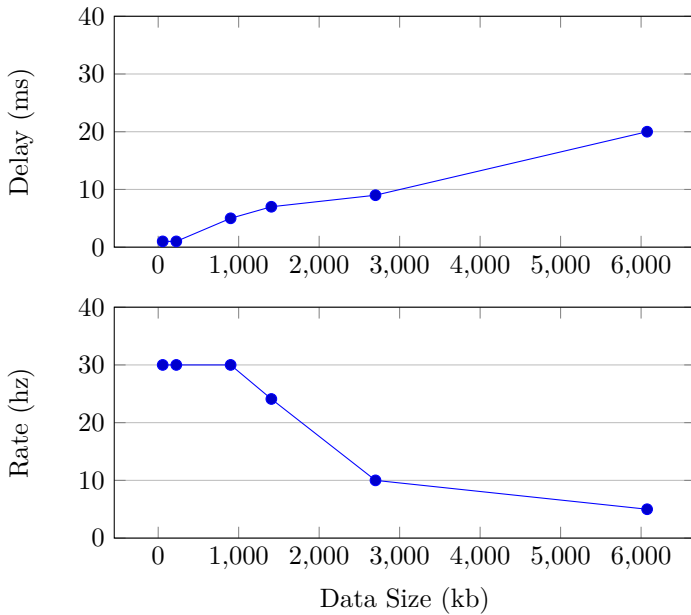
Tabel 4.8: Hasil *delay* dan frekuensi dari pengiriman citra kamera pada *real robot*.

| Width | Height | Size (kb) | Delay (ms) | Rate (hz) |
|-------|--------|-----------|------------|-----------|
| 160 | 120 | 56 | 1.0 | 30.0 |
| 320 | 240 | 225 | 1.0 | 30.0 |
| 640 | 480 | 900 | 5.0 | 30.0 |
| 800 | 600 | 1406 | 7.0 | 24.1 |
| 1280 | 720 | 2700 | 9.0 | 10.0 |
| 1920 | 1080 | 6075 | 20.0 | 5.0 |

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.2.3 Pengujian Pengiriman Citra Kamera Antar-perangkat di Simulasi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan

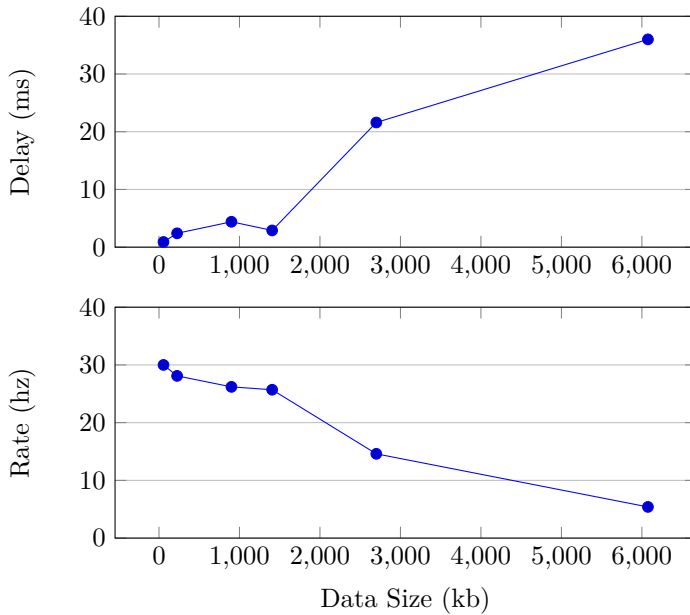


Gambar 4.10: Grafik *delay* dan frekuensi dari pengiriman citra pada *real robot*.

Tabel 4.9: Hasil *delay* dan frekuensi dari pengiriman citra kamera antar-perangkat di simulasi.

| Width | Height | Size (kb) | Delay (ms) | Rate (hz) |
|-------|--------|-----------|------------|-----------|
| 160 | 120 | 56 | 0.9 | 30.0 |
| 320 | 240 | 225 | 2.4 | 28.1 |
| 640 | 480 | 900 | 4.4 | 26.2 |
| 800 | 600 | 1406 | 2.9 | 25.7 |
| 1280 | 720 | 2700 | 21.6 | 14.6 |
| 1920 | 1080 | 6075 | 36.0 | 5.4 |

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris

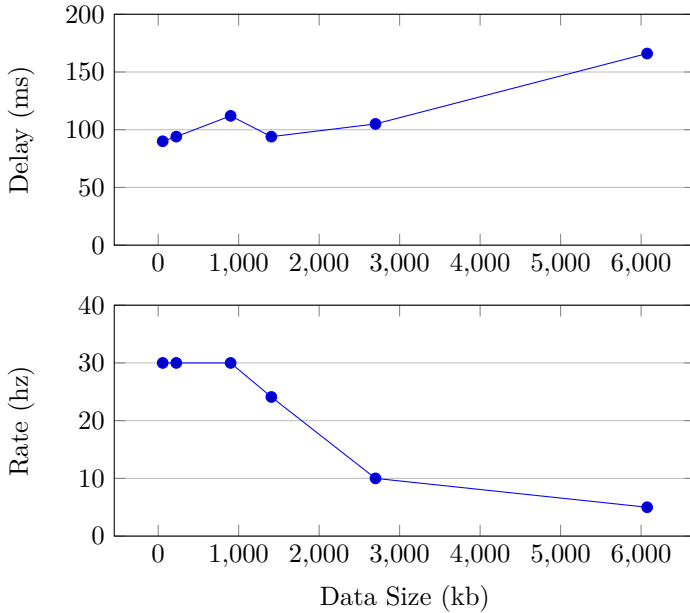


Gambar 4.11: Grafik *delay* dan frekuensi dari pengiriman citra antar-perangkat di simulasi.

Tabel 4.10: Hasil *delay* dan frekuensi dari pengiriman citra kamera antar-perangkat pada *real robot*.

| Width | Height | Size (kb) | Delay (ms) | Rate (hz) |
|-------|--------|-----------|------------|-----------|
| 160 | 120 | 56 | 90.0 | 30.0 |
| 320 | 240 | 225 | 94.0 | 30.0 |
| 640 | 480 | 900 | 112.0 | 30.0 |
| 800 | 600 | 1406 | 94.0 | 24.1 |
| 1280 | 720 | 2700 | 105.0 | 10.0 |
| 1920 | 1080 | 6075 | 166.0 | 5.0 |

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

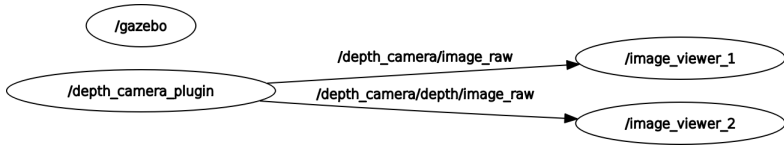


Gambar 4.12: Grafik *delay* dan frekuensi dari pengiriman citra antar-perangkat pada *real robot*.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.3.1 Pengujian Citra *Depth Camera* di Simulasi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Cu-



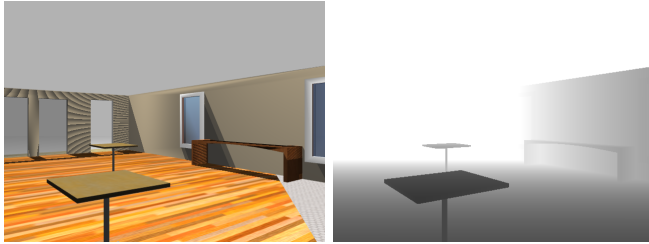
Gambar 4.13: Relasi antar-*node* dari pengujian citra *depth camera* di simulasi.

rabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.3.2 Pengujian Citra *Depth Camera* pada *Real Robot*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu ne-



Gambar 4.14: Perbandingan hasil tangkapan citra berwarna dan citra kedalaman di simulasi.



Gambar 4.15: Relasi antar-*node* dari pengujian citra *depth camera* pada *real robot*.

que. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam

tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



Gambar 4.16: Perbandingan hasil tangkapan citra berwarna dan citra kedalaman pada *real robot*.

4.3.3 Pengujian *Point Cloud* dari Citra *Depth Camera* di Simulasi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus

et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.3.4 Pengujian *Point Cloud* dari Citra *Depth Camera* di Robot

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.4 Pengujian Klasifikasi Pose

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et ma-

lesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.4.1 Pengujian Klasifikasi Pose di Simulasi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan

bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.4.2 Pengujian Klasifikasi Pose di Robot

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.5 Pengujian SLAM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget,

consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.5.1 Pengujian SLAM di Simulasi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.5.2 Pengujian SLAM di Robot

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pembuatan Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.
3. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

5.2 Saran

Untuk pengembangan lebih lanjut pada penelitian tugas akhir ini, terdapat beberapa beberapa saran yang dapat dilakukan, antara lain:

1. Memperbaiki Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.
3. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] M. Heerink, B. Kröse, V. Evers, and B. Wielinga. Assessing acceptance of assistive social agent technology by older adults: the almere model. In *International Journal of Social Robotics*, page 361–375, 2010. doi: 10.1007/s12369-010-0068-5.
- [2] Ros 2 nodes, 2021. URL <https://index.ros.org/doc/ros2/Tutorials/Understanding-RoS2-Nodes/>.
- [3] P. Gonçalves, P. Torres, C. Alves, F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klapacz, S. Magnenat, J. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 1, 2009.
- [4] M. Blatnický, J. Dižo, J. Gerlici, M. Sága, T. Lack, and E. Kuba. Design of a robotic manipulator for handling products of automotive industry. *International Journal of Advanced Robotic Systems*, 17(1), 2020. doi: 10.1177/1729881420906290.
- [5] D. Feil-Seifer and M. J. Mataric. Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 465–468, 2005. doi: 10.1109/ICORR.2005.1501143.
- [6] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp. Assistive gym: A physics simulation framework for assistive robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10169–10176, 2020. doi: 10.1109/ICRA40945.2020.9197411.
- [7] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache. Simulation environment for mobile robots testing using ros and gazebo. In *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 96–101, 2016. doi: 10.1109/ICSTCC.2016.7790647.

- [8] O. Michel. Cyberbotics ltd. webots™: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5, 2004. doi: 10.5772/5618.
- [9] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004. doi: 10.1109/IROS.2004.1389727.
- [10] E. Rohmer, S. P. N. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, 2013. doi: 10.1109/IROS.2013.6696520.
- [11] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [12] C. Rich and C. L. Sidner. Robots and avatars as hosts, advisors, companions, and jesters. *AI Magazine*, 30(1):29, 2009. doi: 10.1609/aimag.v30i1.2765.
- [13] B. Gerkey, R. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. *Proceedings of the International Conference on Advanced Robotics*, 2003.
- [14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. *ICRA Workshop on Open Source Software*, 3, 2009.
- [15] Y. Maruyama, S. Kato, and T. Azumi. Exploring the performance of ros2. In *2016 International Conference on Embedded Software (EMSOFT)*, pages 1–10, 2016. doi: 10.1145/2968478.2968502.
- [16] G. Pardo-Castellote. Omg data-distribution service: architectural overview. In *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, pages 200–206, 2003. doi: 10.1109/ICDCSW.2003.1203555.

- [17] J. M. Schlesselman, G. Pardo-Castellote, and B. Farabaugh. Omg data-distribution service (dds): architectural update. In *IEEE MILCOM 2004. Military Communications Conference, 2004.*, volume 2, pages 961–967 Vol. 2, 2004. doi: 10.1109/MILCOM.2004.1494965.
- [18] R. Dikairono, Setiawardhana, D. Purwanto, and T. A. Sardjono. Cnn-based self localization using visual modelling of a gyro-compass line mark and omni-vision image for a wheeled soccer robot application. *International Journal of Intelligent Engineering and Systems*, 2020. doi: 10.22266/ijies2020.1231.39.
- [19] Muhtadin, Raden Marwan Zanuar, I Ketut Eddy Purnama, and Mauridhi Hery Purnomo. Autonomous navigation and obstacle avoidance for service robot. In *2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pages 1–8, 2019. doi: 10.1109/CENIM48368.2019.8973360.
- [20] M. Muhtadin, M. Arrazi, S. Ali, T. Pratama, D. Wicaksono, A. Putra, I. Ari, A. Maulana, O. Bramastyo, S. Asshakina, M. Attamimi, M. Arifin, M. Hery Purnomo, and D. Purwanto. Ichiro robots winning robocup 2018 humanoid teensize soccer competitions. In *RoboCup 2018: Robot World Cup XXII*, pages 425–435, 2019. doi: 10.1007/978-3-030-27544-0_35.
- [21] Z. Mohamed and G. Capi. Development of a new mobile humanoid robot for assisting elderly people. *Procedia Engineering*, 41:345 – 351, 2012. doi: 10.1016/j.proeng.2012.07.183.

[Halaman ini sengaja dikosongkan]