INVOTEK
Jurnal Inovasi Vokasional dan Teknologi

Volume 20 Number 3, 2020
ISSN: 1411 – 3411 (p) ISSN: 2549 – 9815 (e)
DOI : 10.24036/invotek.v20i3.854

Penerapan Algoritma ORB SLAM-2 Pada Sistem Pemetaan Lingkungan Multi Robot

P. Anggraeni[1], Ridwan[1] dan M.T.A. Asshydiqi[1]
[1]Program Studi Teknologi Rekayasa Otomasi,
Jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung
pipit_anggraeni@polman-bandung.ac.id, ridwan@polman-bandung.ac.id,
taufiqasshydiqi@gmail.com

Abstrak— Sistem multi-robot telah diterapkan pada tugas-tugas kompleks yang biasanya dilakukan oleh manusia. Untuk dapat menjalankan tugasnya, robot perlu bernavigasi ke dari suatu posisi ke posisi lain. Agar dapat bernavigasi dengan baik, robot memerlukan peta sebagai acuannya dalam bernavigasi. *Simultaneous Localization and Mapping* (SLAM) merupakan sebuah metode bagi robot untuk dapat membuat peta dan melakukan lokalisasi. ORB SLAM-2 merupakan sebuah metode SLAM berbasis sensor visual yang kompatibel terhadap kamera monokular, stereo, maupun RGBD. Dengan menggunakan kamera monokular, penelitian ini bertujuan untuk membuat rancangan sistem pemetaan lingkungan multi-robot dengan menggunakan algoritma ORB SLAM-2. Tugas akhir ini merancang sistem desentralisasi sehingga algoritma dijalankan pada kedua robot. Kemudian setiap robot melakukan pemetaan lingkungannya dan mengirimkannya ke komputer agar dapat divisualisasi. Pada percobaannya, rancangan ini berhasil membuat sistem melaksanakan tugasnya dengan baik. Peta yang dihasilkan oleh sistem ini memiliki skala sekitar 1 : 5,81. Sistem juga dapat memvisualisasikan peta yang dihasilkan oleh masing-masing robot pada sebuah komputer server. Berdasarkan hasil percobaan, dapat disimpulkan bahwa sistem pemetaan lingkungan multi-robot menggunakan ORB SLAM-2 dapat dilakukan dengan mendesentralisasi sistem. Dengan ini, beban kerja sistem terbagi menjadi : 1. pemrosesan gambar dilakukan oleh masing-masing robot hingga menghasilkan titik-titik peta; 2. komputer server bertugas untuk memvisualisasikan titik-titik peta yang dihasilkan robot pada antarmuka pengguna.

Kata Kunci : multi-robot, monokular ,VSLAM, SLAM

Abstract— Multi-robot systems have been applied to complex tasks normally performed by humans. To be able to carry out its duties, the robot needs to navigate from position to position. In order to navigate properly, the robot needs a map as a reference in its navigation system. Simultaneous Localization and Mapping (SLAM) is a method for robots to be able to create maps and perform localization. ORB SLAM-2 is a visual sensor-based SLAM method that is compatible with monocular, stereo, and RGBD cameras. By using monocular cameras, this study aims to design a multi-robot environment mapping system using the ORB SLAM-2 algorithm. This study designs a decentralized system so that the algorithm is run on both robots. Then each robot performs a mapping of its environment and sends it to a computer so that it can be visualized. In experiments, this design succeeded in making the system perform its job properly. The map produced by this system has a scale of about 1: 5.81. The system can also visualize the map generated by each robot on a server computer. Based on the experimental results, it can be concluded that a multi-robot environmental mapping system using ORB SLAM-2 can be done by decentralizing the system. With this, the workload of the system is divided into: 1. image processing is carried out by each robot to produce map points; 2. The server computer is tasked with visualizing map points generated by the robot on the user interface.

Keywords : multi-robot, monocular ,VSLAM, SLAM

Received date 22/10/2020, Revised date 29/10/2020, Accepted date 29/10/2020

I. PENDAHULUAN

Smart factory adalah salah satu kunci utama di Industri 4.0 yang dianggap sebagai generasi industri berikutnya [1], [2]. Pada industri 4.0, perusahaan manufaktur menjadi lebih fleksibel, terukur, dan kompatibel [3]. *Mobile robot* adalah robot penanganan logistik yang fleksibel dan cerdas, yang mampu melakukan kemudi otomatis untuk melakukan berbagai fungsi sebagai alat transfer logistik [4]. Dalam pendekatan untuk mencapai efisiensi yang lebih tinggi, penggunaan beberapa robot disarankan dibandingkan sistem robot tunggal [5], [6].

Sistem multi-robot telah diterapkan pada tugas-tugas kompleks yang biasanya dilakukan oleh manusia. Tugas-tugas tersebut contohnya pemadaman kebakaran, deteksi ranjau darat, dekontaminasi radiasi, pekerjaan pertanian, konstruksi, misi bawah air, pekerjaan gudang, dan *Search and Rescue (SAR)*. Fenomena ini terjadi karena penggunaan beberapa robot untuk melaksanakan tugas-tugas tersebut dinilai dapat meningkatkan ketahanan dan efisiensi dibandingkan dengan penggunaan satu robot [7]. Namun, beberapa pekerjaan sistem multi-robot otonom adalah masalah yang sulit di bidang otomatisasi cerdas [8]. Salah satu masalahnya adalah diperlukannya kemampuan dari robot-robot untuk menciptakan peta dari lingkungan tempatnya bekerja yang kemudian digunakan sebagai acuan dalam bernavigasi [9].

Simultaneous Localization and Mapping (SLAM) adalah salah satu masalah paling mendasar, namun paling menantang dalam *mobile robot*. SLAM mendefinisikan masalah robot agar dapat mencapai otonomi penuh, yaitu robot harus memiliki kemampuan untuk menjelajahi lingkungannya tanpa campur tangan pengguna, membuat peta yang andal, dan melakukan lokalisasi dirinya sendiri terhadap peta. Secara khusus, jika data sensor pemosisian global (GPS) dan *beacon* eksternal tidak tersedia, robot bagaimanapun caranya harus tetap, dengan sendirinya, menentukan titik referensi yang sesuai untuk membangun peta [10]. Disisi lain, penggunaan kamera sebagai sensor dalam SLAM berbasis visual yang juga dikenal sebagai VSLAM semakin diminati peneliti karena kaya akan informasi visual yang tersedia dari sensor video. VSLAM menghasilkan beberapa solusi dengan menggunakan beberapa tipe sensor kamera yang berbeda seperti monokular, stereo, *omni-directional*, *time of flight*, dan RGB-D (kombinasi antara warna dan kedalaman) [11]. Salah satunya adalah ORB SLAM-2 [13] yang kompatibel terhadap kamera monokular, stereo, maupun RGB-D. Pada penelitiannya algoritma ini berhasil untuk melakukan relokalisasi, penutupan lup, dan menggunakan kembali peta yang telah dibuat dengan akurasi yang memuaskan.

Dari uraian di atas dapat terlihat bahwa *mobile robot* telah menjadi bagian vital di kehidupan manusia. *Mobile robot* dapat menggantikan peran manusia untuk menyelesaikan pekerjaan yang berbahaya, kotor, monoton, hingga pekerjaan yang sulit untuk diselesaikan oleh manusia. Dalam hal pendekatan efisiensi, sistem multi-robot otonom lebih disarankan dalam penggunaan *mobile robot*. Pada sistem ini, robot-robot memerlukan sebuah peta global yang digunakan sebagai acuan robot dalam bernavigasi untuk menjalankan tugasnya.

Permasalahan yang akan dihadapi adalah rancangan sistem pemetaan lingkungan multi-robot dengan menggunakan algoritma ORB SLAM dengan batasan permasalahan:

a) Jumlah robot yang digunakan pada sistem multi-robot ini adalah 2 robot.
b) Robot yang digunakan adalah Festo Robotino 3 yang tertanamkan komputer dengan prosesor *dual core* 2.4 GHz dan 8 GB RAM.
c) Pemetaan dilakukan dengan hanya menggunakan sensor kamera berjenis monokular tanpa dibantu sensor lain. Kamera yang digunakan memiliki resolusi dan memiliki kamera dengan resolusi maksimal hingga 1920x1080 piksel dengan banyaknya frame tiap detik sebesar 30 fps, memiliki autofokus dan sudut pandang sebesar 78º.
d) Sistem multi-robot yang dibuat tidak meliputi penggabungan peta *(map merging)*.
e) Kondisi lingkungan pada saat proses pemetaan harus statis atau konsisten, yaitu tidak ada perubahan posisi maupun orientasi objek.
f) Maksimal kecepatan liner robot adalah 1 m/s, sedangkan maksimal kecepatan angular robot adalah 0,3 rad/s.

Tujuan dari penelitian ini adalah mengembangkan hasil penelitian terdahulu yang telah menghasilkan algoritma ORB SLAM-2 dengan cara merancang ulang algoritma tersebut agar dapat diaplikasikan pada sistem pemetaan lingkungan multi-robot secara waktu riil dengan segala keterbatasan perangkat keras yang digunakan. Kemudian peta hasil pemetaan tersebut dievaluasi, dibandingkan dengan keadaan nyata

dari lingkungan robot. Hasil dari tugas akhir ini diharapkan dapat digunakan sebagai modal untuk pembuatan sebuah peta global yang dihasilkan oleh beberapa robot dengan menggunakan metode *map merging*.

II. TINJAUAN PUSTAKA

Pada tahun 2017, penelitian [14] membandingkan 4 algoritma SLAM berbasis sensor visual kamera monokular terbaru pada saat itu, yaitu ORB SLAM, REMODE, LSD-SLAM, dan DPPTAM. Penelitian dilakukan dengan memberikan masukan video ke masing-masing algoritma. Penelitian ini menyatakan keempat algoritma tersebut dinilai baik dalam mendeteksi objek volumetrik, sudut, rintangan, dan fitur lainnya. Namun, algoritma-algoritma tersebut mengalami kesulitan saat berusaha mendeteksi dinding yang berwarna homogen sehingga algoritma tadi tidak dapat merekonstruksinya.

Penelitian [13] mengenalkan sebuah *open-source* SLAM sistem pertama untuk kamera monokular, stereo, maupun RGB-D. Sistem ini memiliki 3 algoritma utama yang disebut pelacakan *(tracking)*, pemetaan lokal *(local mapping)*, dan penutupan lup *(loop closing)*. Pada percobaannya, algoritma ini diuji dengan 3 buah *datasets* popular yaitu KITTI, EuRoC, dan TUM RGB-D. Hasilnya, algoritma ini mampu untuk melakukan relokalisasi, penutupan lup, dan menggunakan kembali peta yang telah dibuat secara waktu riil pada sebuah komputer dengan CPU Intel Core i7-4790 dengan akurasi yang memuaskan. Namun penelitian ini hanya dapat membuat peta dari sebuah kamera.

Penelitian [15] berhasil melakukan komunikasi antara beberapa robot melalui jaringan nirkabel. Juga, memungkinkan untuk menggunakan konfigurasi berbiaya rendah ini untuk memantau dan mengendalikan beberapa robot dengan hanya menggunakan satu workstation. Penelitian ini menggunakan robot MiniLab Enova sebagai agennya. Skema kontrol komunikasi pada penelitian ini dikembangkan dengan menggunakan ROS multi-master.

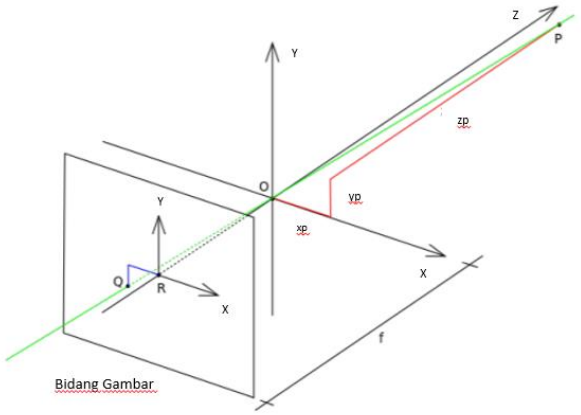
Penelitian [16] berhasil menggabungkan 3 buah peta yang dihasilkan dari 3 buah UAV yang masing-masing dilengkapi sebuah kamera monocular menjadi sebuah peta global secara waktu riil. Pada penelitian ini, gambar yang dihasilkan dari setiap kamera diproses oleh masing masing agen sehingga menghasilkan *keyframes* dan *map points* (titik peta), yang kemudian dikirimkan ke server untuk diolah kembali. Pada server inilah proses yang berat dilakukan yaitu manajemen peta, pengenalan tempat, penggabungan peta, dan global *bundle adjustment*. Namun pada penelitian ini terdapat kelemahan yaitu jika sistem mengalami kegagalan pelacakan *(tracking)*, proses pembuatan peta tidak dapat dilanjutkan melainkan harus mengulangi proses pembuatan peta tersebut dari awal.

A. Robotino

Robotino adalah platform *mobile robot* khususnya *Unmanned Ground Vehicle (UGV)* untuk penelitian dan pendidikan. Robotino dikendalikan oleh sistem PC standar industri yang cukup kuat untuk merencanakan rute untuk berjalan otomatis seutuhnya. Melalui WLAN, robotino dapat mengirim semua data pembacaan sensor ke PC eksternal. Sementara itu, perintah kontrol pun dapat dikeluarkan oleh PC eksternal. Dengan cara ini, program kontrol dapat berjalan di PC eksternal atau di robotino secara langsung. Mode campuran atau kontrol bersama juga memungkinkan untuk dilakukan. Dengan *drive omnidirectional*, sensor, antarmuka, dan ekstensi spesifik aplikasi, robotino dapat digunakan dengan sangat fleksibel, juga dengan roda omninya memungkinkan robotino bergerak lebih bebas dan mudah [17].

B. Kamera

Untuk dapat memproyeksikan koordinat dunia 3D ke dalam bidang gambar, diperlukan sebuah model untuk mencapai tujuan ini. Model yang paling umum digunakan adalah *pinhole camera model* yang berasumsi bahwa cahaya yang mengenai bidang gambar hanya melalui titik lubang bukaan kecil pada penghalang pada posisi O yang disebut juga sebagai titik tengah kamera. Cahaya yang melewati lubang ini dan kemudian diproyeksikan pada bidang gambar. Gambar ditempatkan pada jarak f dari O di mana *f* merupakan *focal length* dari kamera. Jika kita menganggap sistem koordinat dunia 3D dengan titik pusat di O dan arah sumbu *z* searah dengan arah pandangan kamera serta sumbu *x* dan *y* yang mencakup bidang gambar. Pertimbangkan juga titik P = (x_p, y_p, z_p) dalam sistem koordinat ini. Jika mempertimbangkan sistem koordinat 2D dalam bidang gambar dengan pusat di titik R yang merupakan persimpangan sumbu *z* dari sistem koordinat dunia 3D dan titik Q = (X_q, Y_q) di dalam bidang gambar. Gambar yang menunjukkan konfigurasi ini dapat dilihat pada Gambar 1.



Gambar 1. Konfigurasi koordinat kamera dan koordinat dunia [20].

Untuk dapat memproyeksikan koordinat dunia 3D ke dalam bidang gambar, diperlukan sebuah model untuk mencapai tujuan ini. Model yang paling umum digunakan adalah *pinhole camera model* yang berasumsi bahwa cahaya yang mengenai bidang gambar hanya melalui titik lubang bukaan kecil pada penghalang pada posisi O yang disebut juga sebagai titik tengah kamera. Cahaya yang melewati lubang ini dan kemudian diproyeksikan pada bidang gambar. Gambar ditempatkan pada jarak f dari O di mana f merupakan *focal length* dari kamera. Jika kita menganggap sistem koordinat dunia 3D dengan titik pusat di O dan arah sumbu z searah dengan arah pandangan kamera serta sumbu x dan y yang mencakup bidang gambar. Pertimbangkan juga titik P = (x_p, y_p, z_p) dalam sistem koordinat ini. Jika mempertimbangkan sistem koordinat 2D dalam bidang gambar dengan pusat di titik R yang merupakan persimpangan sumbu z dari sistem koordinat dunia 3D dan titik Q = (X_q, Y_q) di dalam bidang gambar. Gambar yang menunjukkan konfigurasi ini dapat dilihat pada Gambar 1. Dengan model ini transformasi dari P dan Q dapat didefinisikan dengan mengamati bahwa segitiga terbentuk antara Q, pusat bidang bayangan dan O. Segitiga yang berlawanan terbentuk antara P, proyeksi P pada z dan O. Dengan menggunakan hukum segitiga sebangun, maka didapatkan hubungan berikut:

$$Q = \begin{pmatrix} X_q \\ Y_q \end{pmatrix} = \frac{f}{z_p} \begin{pmatrix} x_p \\ y_p \end{pmatrix} \quad (1)$$

Namun karena gambar yang diperoleh dari kamera adalah digital, dan diubah menjadi bentuk piksel. Ini memunculkan beberapa parameter tambahan dan agak memperumit transformasi. Masalah pertama adalah bahwa gambar digital memiliki titik pusat yang ditentukan di sudut kiri atas sehingga terdapat *offset* antara proyeksi dan gambar digital yang diwakili oleh vektor translasi (c_x, c_y). Selain hubungan jarak antara piksel dan jarak yang sama pada proyeksi bidang gambar perlu didefinisikan. Oleh karena itu f perlu diganti dengan f_x dan f_y dalam relasi di atas. Jadi dalam gambar digital, hubungan di atas dapat ditulis sebagai:

$$Q = \begin{pmatrix} X_q \\ Y_q \end{pmatrix} = \begin{pmatrix} f_x \frac{x_p}{z_p} + c_x \\ f_y \frac{y_p}{z_p} + c_y \end{pmatrix} \quad (2)$$

Untuk menulis persamaan ini dengan lebih ringkas, dapat digunakan sistem koordinat homogen, sehingga persamaan menjadi:

$$Q = \begin{pmatrix} f_x x_p + c_x z_p \\ f_y y_p + c_y z_p \\ z_p \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = MP \quad (3)$$

Untuk mendapatkan matriks kamera K, maka persamaan diuraikan menjadi:

$$Q = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} (I \quad 0)P = K(I \quad 0)P \quad (4)$$

Sehingga matriks K memberikan semua informasi yang diperlukan untuk melakukan transformasi koordinat dunia ke koordinat piksel kamera. Ini juga disebut matriks kalibrasi yang juga dikenal sebagai parameter intrinsik kamera [20], [21].

Namun, pada kasus kamera yang bergerak, titik bukaan kamera tidak dapat dijadikan sebagai titik pusat koordinat dunia, sehingga dibutuhkan transformasi dari koordinat dunia ke koordinat kamera. Untuk itu dibutuhkan matriks rotasi R dan vektor translasi t. Ini dapat dirumuskan menjadi:

$$P = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} P_w \quad (5)$$

Dimana P_w adalah titik di koordinat dunia. Dengan mensubtitusikan P pada persamaan ini, maka akan didapatkan:

$$Q = K(R \quad t)P_w \quad (6)$$

Matriks R dan t memiliki peran yang sangat penting pada sistem visoal odometri maupun visual SLAM, karena posisi dan orientasi (C_w) dari kamera pada koordinat dunia dapat didefinisikan dengan:

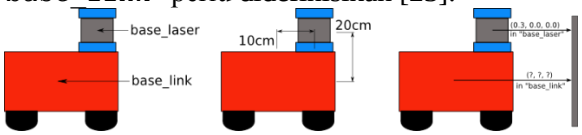
$$C_w = -R^T t \quad (7)$$

C. Robot Operating System (ROS)

ROS adalah peranti tengah terbuka dioperasikan melalui sistem operasi Linux yang digunakan untuk kendali masukan dan keluaran serta komunikasi robot. ROS mampu memproses data dan menyediakan *package* yang bertanggung jawab sebagai pustaka robot. ROS memungkinkan pengendalian perangkat melalui *Nodes* yang

tersedia pada pustaka robot. *Nodes* dapat mengirim atau menerima data berupa pesan terhadap suatu *Topics*. *Topics* ini mengendalikan aktuator dan sensor pada robot secara langsung dengan cara mengirimkan pesan dari *Nodes*. *Nodes* dapat berupa program yang tersedia dalam bahasa C++ dan Python [22].

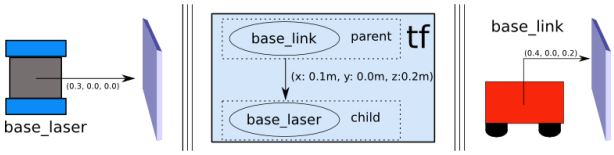
Agar sensor pada robot dapat digunakan dengan benar, sistem harus dapat mengetahui dimana posisi titik pusat sensor "`base_laser`" tersebut diletakan mengacu terhadap titik pusat robot itu sendiri yang disebut "`base_link`". Sebagai contoh ada sebuah *mobile robot* dengan sebuah sensor laser diatasnya seperti pada Gambar 2. Asumsikan bahwa ada beberapa data sensor laser dalam bentuk jarak dari titik pusat laser. Dengan kata lain, ada beberapa data dalam bingkai koordinat "`base_laser`". Agar robot ini dapat menghindari rintangan, data yang diterima dari bingkai "`base_laser`" perlu diubah menjadi bingkai "`base_link`", dengan kata lain hubungan antara frame koordinat "`base_laser`" dan "`base_link`" perlu didefinisikan [23].



Gambar 2. Ilustrasi sebuah mobile robot dengan sebuah sensor laser [23].

Dalam mendefinisikan hubungan ini, perlu diketahui bahwa laser dipasang 10 cm ke depan dan 20 cm di atas titik pusat robot. Ini akan memberikan offset translasi yang menghubungkan frame "`base_link`" ke frame "`base_laser`". Secara khusus, dapat diketahui bahwa untuk mendapatkan data dari bingkai "`base_link`" ke bingkai "`base_laser`" diperlukan translasi (x: 0,1 m, y: 0,0 m, z: 0,2 m), dan untuk mendapatkan data dari bingkai "`base_laser`" ke bingkai "`base_link`" diperlukan translasi yang berlawanan (x: -0.1 m, y: 0.0 m, z: -0.20 m) [23].

Menerjemahkan hubungan tersebut dapat dilakukan dengan mudah tanpa menggunakan tf, namun hal ini akan menjadi menyebalkan ketika jumlah bingkai koordinat semakin banyak. Hal ini dapat diselesaikan dengan mudah dengan mendefinisikan dan menyimpan hubungan antara bingkai "`base_link`" dan "`base_laser`" menggunakan tf. Untuk dapat menggunakan tf diperlukan sebuah pohon transformasi. Secara konseptual, setiap *node* di pohon transformasi sesuai dengan bingkai koordinat dan setiap tepi sesuai dengan transformasi yang perlu diterapkan untuk bergerak dari *node* saat ini ke anaknya. Tf menggunakan struktur pohon untuk menjamin bahwa hanya ada satu traversal tunggal yang menghubungkan dua bingkai koordinat bersama-sama, dan mengasumsikan bahwa semua tepi dalam pohon diarahkan dari *node* induk ke anak [23]. Hal ini seperti yang digambarkan pada Gambar 3.



Gambar 3. Pohon transformasi dari sebuah robot dengan sensor laser [23].

Untuk membuat pohon transformasi dari contoh di atas, dibuat dua *node*, satu untuk bingkai koordinat "`base_link`" dan satu untuk bingkai koordinat "`base_laser`". Untuk membuat batas di antaranya, pertama-tama diperlukan untuk memutuskan simpul mana yang akan menjadi induk dan yang akan menjadi anak. Perbedaan ini penting karena mengasumsikan bahwa semua transformasi berpindah dari induk ke anak. Sebagai contoh, bingkai koordinat "`base_link`" dipilih sebagai induk karena ketika potongan/sensor lain ditambahkan ke robot, akan lebih masuk akal bagi mereka untuk menghubungkannya ke bingkai "`base_laser`" dengan menelusuri melalui bingkai "`base_link`". Ini berarti transformasi yang terkait dengan tepi yang menghubungkan "`base_link`" dan "`base_laser`" harus (x: 0,1 m, y: 0,0 m, z: 0,2 m). Dengan susunan transformasi ini, mengonversi pemindaian laser yang diterima dalam bingkai "`base_laser`" ke bingkai "`base_link`" semudah membuat panggilan ke *library* tf [23].

III. METODE

Pada bagian ini akan dibahas mengenai metode penyelesaian masalah pada penelitian ini yang meliputi arsitektur sistem yang dibuat, perangkat yang digunakan. Selain itu juga pada bagian ini akan dijelaskan diagram alir sistem serta algoritma yang digunakan yaitu ORB SLAM-2.

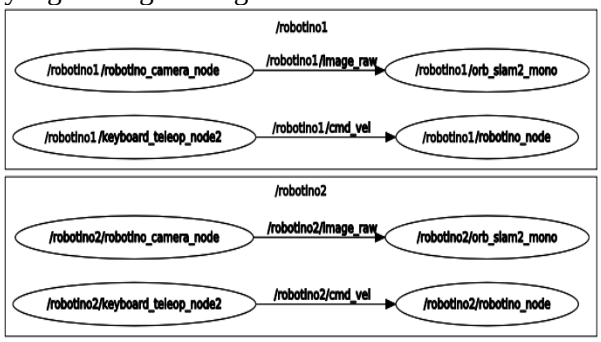
A. Arsitektur Sistem

Sistem ini dirancang untuk dapat menampilkan peta yang dihasilkan oleh 2 buah kamera yang disematkan pada 2 unit mobile robot. Robot-robot bertugas untuk membawa kamera tersebut mengeksplorasi lingkungannya agar kamera mendapatkan gambar dari lingkungannya.

Gambar-gambar tersebut kemudian dikomputasi oleh robot hingga menghasilkan titik-titik peta dari lingkungan tersebut. Titik peta ini kemudian dikirimkan ke komputer server untuk digunakan sebagai pembuat peta maupun sebagai acuan lokalisasi. Seperti yang dijelaskan pada Gambar 4, antarmuka yang digunakan untuk mengirim gambar ke robot menggunakan USB, sedangkan antarmuka komunikasi antara robot dengan komputer menggunakna jaringan nirkabel.

Gambar 4. Arsitektur sistem yang dibangun.

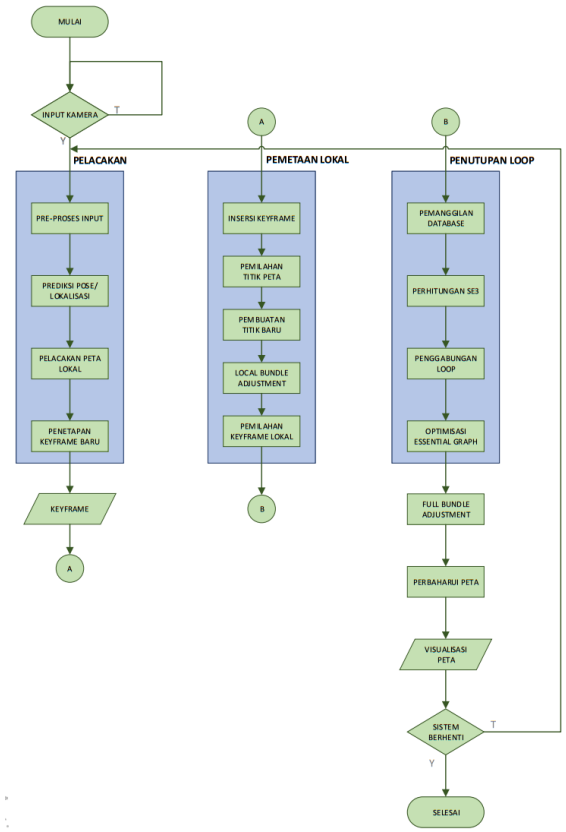
Sistem yang dibuat pada tugas akhir ini menggunakan ROS sebagai *middleware*, yaitu suatu lapisan perangkat lunak yang berada di antara *router* dengan kontroler. Gambar 5 menggambarkan konfigurasi *nodes* dan *topics* yang terbentuk pada sistem ini, sistem yang dibentuk pada tugas akhir ini dibangun oleh 2 *Namespace* yang masing-masing terdiri dari 4 *Node*.



Gambar 5 Konfigurasi *node* dan *topic* pada sistem.

Sistem ROS dimulai dengan menginisiasi komputer sebagai master, lalu komputer juga mengeksekusi *Node* `keyboard_teleop` untuk masing-masing robot. Kemudian *Node* `robotino_node`, `orb_slam_2_mono`, dan `robotino_camera_node` dieksekusi di masing-masing robot. *Node* `keyboard_teleop` akan mengirimkan *Topic* bernama `/cmd_vel` ke masing-masing robot melalui *Topic* `/robotino_node`. *Topic* tersebut membawa pesan perintah pergerakan robot baik linier maupun angular sehingga robot dapat bergerak membawa kameranya untuk melakukan eksplorasi lingkungan. Kamera tersebut diperintah oleh `/robotino_camera_node` sehingga dapat mengirimkan data gambarnya dengan *Topic* `/image_raw` ke *Topic* `/orb_slam_2_mono`. Pada *Node* ini gambar tadi dikomputasi sehingga menjadi data yang dibutuhkan untuk membuat peta, seperti titik peta, posisi dan orientasi robot, gambar hasil olahan algoritma, dan *trajectory* robot.

Gambar 6 di bawah ini merupakan paparan diagram alir dari sistem yang telah dibuat.



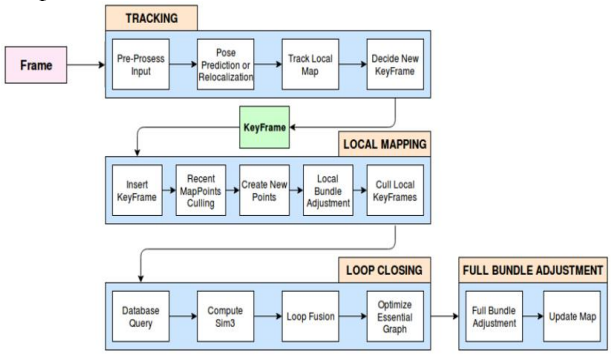
Gambar 6. Diagram alir sistem.

B. Algoritma ORB SLAM-2

Pada dasarnya ORB SLAM-2 menggunakan *ORB (Oriented FAST and Rotated BRIEF)* sebagai *feature detector and descriptor*, yang merupakan *oriented multi-scale FAST corners* dengan deskriptor 256 bit yang terasosiasi. Algoritma ini digunakan karena kecepatannya dalam untuk menghitung dan mencocokkan, sementara mereka memiliki invarian yang baik untuk *viewpoint* [24]. Algoritma ORB merupakan gabungan dari *FAST (Features From Accelerated Segment Test) keypoint detector* dan *BRIEF (Binary Robust Independent Elementary Features) descriptor* dengan beberapa modifikasi untuk meningkatkan

kemampuannya. Pertama-tama, algoritma FAST digunakan untuk mendapatkan *keypoint.* Idenya adalah bahwa jika suatu piksel secara signifikan berbeda dari piksel sekitarnya maka lebih cenderung menjadi titik sudut. Setelah itu, algoritma ORB menggunakan algoritma BRIEF untuk menghitung deskriptor pada setiap titik. BRIEF adalah deskriptor vektor biner yang vektornya terdiri dari angka 0 dan 1 [25].

ORB SLAM-2 adalah lanjutan dari penelitian terdahulu yaitu ORB SLAM yang memungkinkan penggunaan kamera RGB-D atau kamera stereo sebagai ganti kamera monokuler [13]. Sebagai pendahulunya, ORB SLAM menggunakan tiga utas yang bekerja secara paralel. Seperti pada Gambar 7 pertama-tama utas *tracking* atau pelacakan, yang digunakan untuk mencocokkan peta lokal dengan fitur yang diekstraksi untuk setiap *frame*, dan juga untuk meminimalkan galat dari *reprojection* untuk secara bersamaan melokalisasi kamera. Utas kedua yaitu *local mapping* atau pemetaan lokal yang digunakan untuk mengoptimalkan dan mengelola peta lokal melalui *local bundle adjustment*. Utas terakhir yaitu *loop closing* atau penutupan lup bertugas untuk melakukan optimasi *pose-graph* untuk memperbaiki penyimpangan dan mendeteksi lup. Aspek baru lain dalam ORB SLAM-2 adalah bahwa dalam utas penutupan lup, terdapat utas baru yaitu utas keempat yang melakukan *full bundle adjustment* dari seluruh peta untuk mencapai rekonstruksi lingkungan yang dioptimalkan dan konsisten [20].



Gambar 7 Aliran data pada algoritma ORB SLAM-2.

Peta yang dihasilkan oleh ORB SLAM-2 dibangun dari sekumpulan *map point* (titik peta) yang merupakan hasil ekstraksi ORB dari *keyframe*. Setiap titik peta p_i memiliki informasi : posisinya pada ruang 3D $X_{w,i}$; orientasinya n_i ; serta jarak maksimum dan minimum titik tersebut dapat terobservasi oleh kamera. Setiap *keyframe* memiliki informasi : posisi kamera K_i, inilah yang mentransformasi titik peta ke sistem koordinat kamera ; intrinsik kamera, seperti *focal point* dan *principal point* ; seluruh ORB fitur yang diekstraksi, baik yang diasosiasikan ke titik peta maupun tidak [24].

Salah satu aspek penting lain dari ORB SLAM-2 adalah *covisibility-graph*, yang digunakan untuk menghubungkan dua *keyframe* yang memiliki titik observasi yang serupa. Grafik digunakan untuk menentukan lingkungan lokal yang memungkinkan pelacakan dan pemetaan yang bekerja secara lokal. ORB SLAM-2 juga menggunakan *bag of words*, yang merupakan modul pengenalan tempat pada sistem ini. Ini digunakan untuk mendeteksi lup yang terjadi jika sudah ada peta lokal telah didapatkan, peta tersebut menginisialisasi ulang dirinya ketika sistem gagal melacak posisinya [20].

IV. HASIL DAN PEMBAHASAN

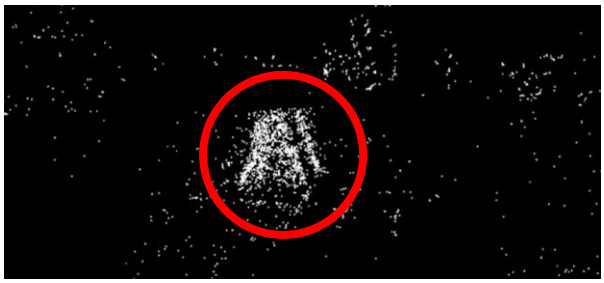
A. Pendeteksian Objek

Sistem diperintahkan untuk dapat mendeteksi objek dengan memproyeksikan objek tersebut ke dalam peta yang dibuat sistem. Pada pelaksanaannya, sistem berhasil membangun titik peta hingga menyerupai bentuk dari profil objek. Berikut merupakan hasil proyeksi objek yang dibangun oleh titik peta dengan beberapa profil objek yang berbeda. Hasil proyeksi objek ditandai dengan lingkaran merah.

1) Objek berprofil huruf A

Pada Gambar 8 di bawah terlihat sekumpulan titik-titik peta yang membentuk profil huruf A. Titik-titik ini memiliki rentang koordinat (0.341, -0.029, -0.132), (0.372, -0.033, 0.006), (0.411, -0.062, 0.006), (0.446, -0.079, -0.137), (0.413 -0.059, -0.133), (0.386, -0.081, -0.078), (0.384, -0.042, -0.081), dan (0.372, -0.046, -0.132).

Gambar 8a. Objek berprofil huruf A.

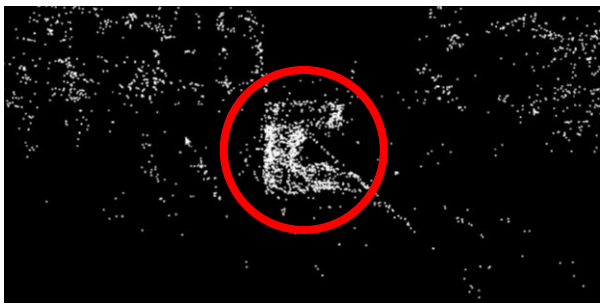


Gambar 8b. Proyeksi objek berprofil huruf A pada peta.

2) Objek berprofil huruf C

Pada Gambar 9 terlihat sekumpulan titik-titik peta yang membentuk profil huruf C. Titik-titik ini memiliki rentang koordinat (0.619, -0.574, 0.017), (0.536, 0.056, 0.016), (0.528, 0.068, -0.180), (0.623, -0.042, -0.189), (0.623 -0.045, -0.140), (0.448, 0.180, -0.131), (0.568, 0.023, -0.032), dan (0.611, -0.042, -0.23).

Gambar 9a. Objek berprofil huruf C.



Gambar 9b. Proyeksi objek berprofil huruf C pada peta.

3) Galon air

Pada Gambar 10 terlihat sekumpulan titik-titik peta yang membentuk profil galon yang memiliki alas berbentuk lingkaran dengan koordinat titik pusat (0.437, 0.122, -0.078) dengan tinggi pada sumbu z dengan rentang -0.078 hingga 0.095.

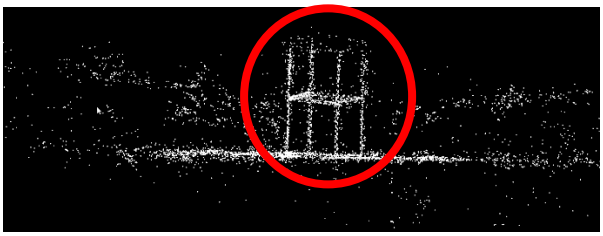
Gambar 10a. Objek galon air.

Gambar 10b. Proyeksi objek galon pada peta.

4) Objek berprofil balok

Pada Gambar 11 terlihat sekumpulan titik-titik peta yang membentuk profil balok. Titik-titik ini memiliki rentang koordinat (0.197, -0.422, -0.104), (0.160, -0.345, -0.106), (0.129, -0.348, 0.059), (0.167, -0.428, 0.066).

Gambar 11a. Objek berprofil balok.

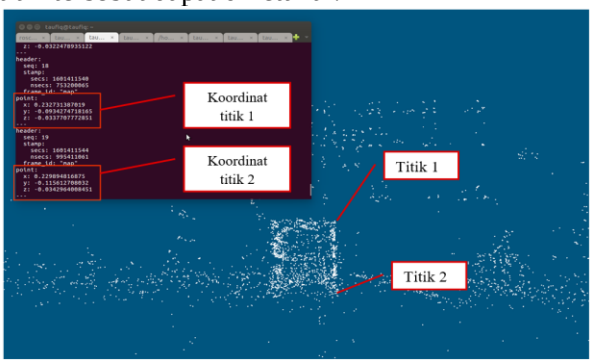


Gambar 11b. Proyeksi objek berprofil balok pada peta.

B. Pengukuran Dimensi Objek pada Peta

Seperti yang sudah disebutkan sebelumnya pengukuran ini dilakukan dengan maksud untuk

mengetahui perbandingan antara dimensi objek sebenarnya dan dimensi pada peta. Sistem membuat peta sebanyak 5 kali, peta ini berisikan sebuah objek yang sama yaitu sebuah objek berbentuk balok yang akan dibandingan dengan dimensi aslinya. Pengukuran dimensi objek sebenarnya dilakukan menggunakan alat ukur dengan ketelitian 1 mm, sedangkan pengukuran dimensi objek pada peta dilakukan dengan menggunakan perangkat lunak Rviz dengan cara mem-*publish* koordinat titik-titik sudut objek tersebut, seperti yang digambarkan pada Gambar 12. Dengan mengetahui koordinat dari 2 titik, maka panjang sebuah garis yang dibentuk oleh kedua titik tersebut dapat diketahui.



Gambar 12. Contoh pengukuran dimensi objek.

Tabel 1 adalah hasil perbandingan antara dimensi objek sebenarnya dan dimensi objek pada peta :

Tabel 1. Hasil pengukuran objek.

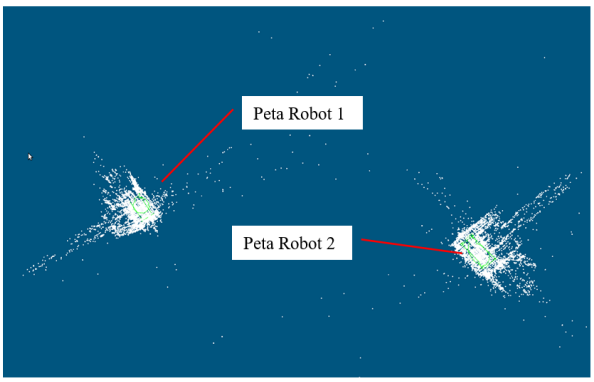
	Sebenarnya	Peta						Sebenarnya/ rata-rata
		1	2	3	4	5	Rata-rata	
Lebar	200	34	35	32	32	32	33.0	6.06
Tinggi	240	42	46	40	41	41	42.0	5.71
Panjang	340	61	64	60	62	53	60.0	5.67
Perbandingan rata-rata								5.81

Berdasarkan Tabel 1 dapat disimpulkan bahwa objek pada peta yang dibuat sistem merupakan representasi dari objek lingkungan robot dengan perbandingan rata-rata 1 : 5.81.

C. Pengujian Pemetaan Multi-Robot

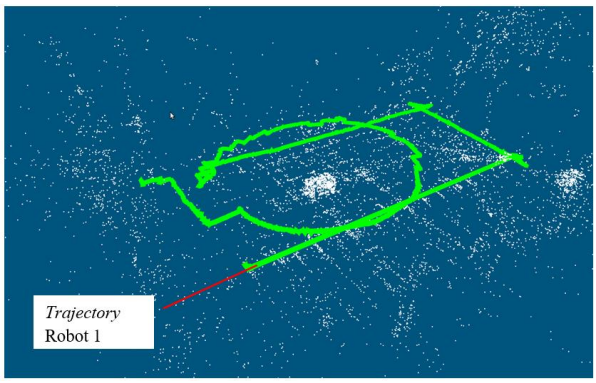
Pengujian dimulai dengan memetakan ruangan dengan sebuah objek di dalamnya seperti yang ditunjukkan pada Gambar 13 menunjukan bahwa sistem berhasil untuk memetakan objek dan ruangan tempat robot-robot bekerja. Garis berwarna hijau pada gambar merupakan *trajectory* atau lintasan yang dilalui robot saat proses pemetaan berlangsung.

Gambar 13a. Lingkungan robot

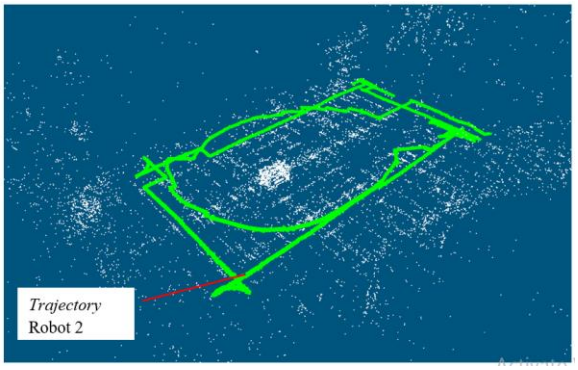


Gambar 13b. Hasil pemetaan sebuah ruangan dengan satu objek.

Apabila diperbesar, maka peta yang dibentuk oleh masing-masing robot akan seperti Gambar 14 berikut:



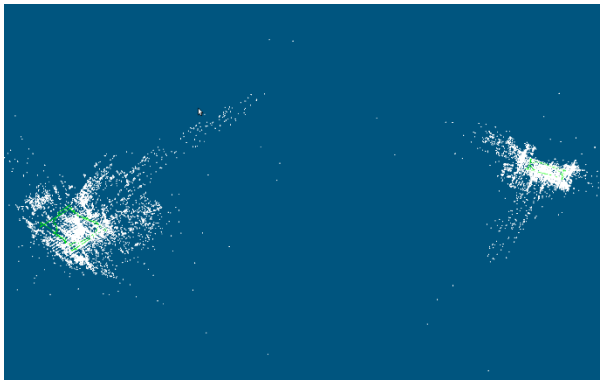
Gambar 14a. Peta yang dihasilkan robot 1



Gambar 14b. Peta yang dihasilkan robot 2.

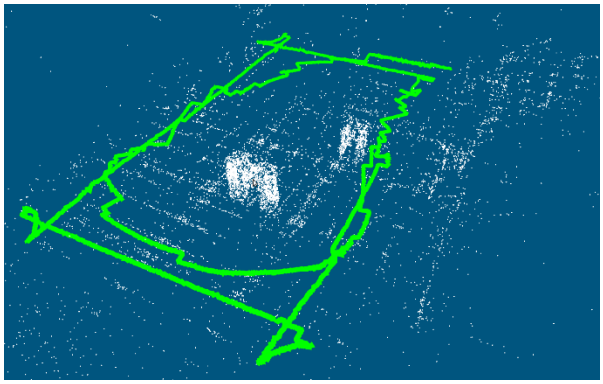
Setelah sistem mampu memetakan lingkungan dengan satu objek, kemudian sistem diuji untuk dapat memetakan lingkungan yang terdapat lebih dari satu objek rintangan seperti pada Gambar 15 berikut:

Gambar 15a. Lingkungan robot dengan dua objek rintangan.

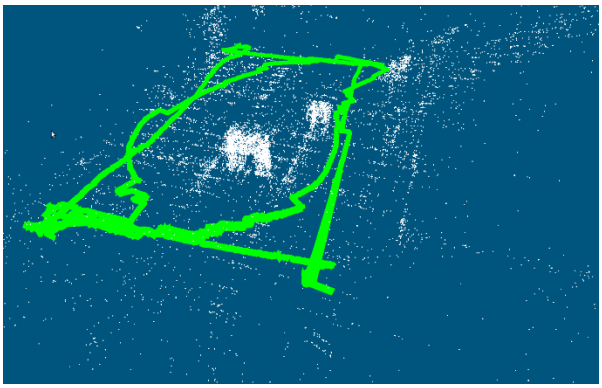


Gambar 15b. Hasil pemetaan dengan 2 objek rintangan.

Apabila diperbesar, maka peta yang dibentuk oleh masing-masing robot akan seperti Gambar 16 berikut:



Gambar 16a. Hasil pemetaan robot 1.



Gambar 16b. Hasil pemetaan robot 2.

V. KESIMPULAN

Berdasarkan hasil dari seluruh pengujian, dapat disimpulkan bahwa sistem pemetaan lingkungan multi-robot dengan menggunakan metode VSLAM dapat dilakukan dengan menggunakan algoritma ORB SLAM-2. Sistem mampu untuk merepresentasikan objek-objek yang ada di lingkungannya ke dalam sebuah peta dengan perbandingan antara dimensi peta dan dimensi sebenarnya 1 : 5,81. Keterbatasan perangkat keras yang digunakan pada tugas akhir ini seperti yang telah disebutkan sebelumnya pada batasan masalah, dapat diatasi dengan mendesentralisasi sistem. Dengan ini, beban kerja sistem terbagi menjadi : 1. pemrosesan gambar dilakukan oleh masing-masing robot hingga menghasilkan titik-titik peta; 2. komputer server bertugas untuk memvisualisasikan titik-titik peta yang dihasilkan robot pada antarmuka pengguna.

DAFTAR PUSTAKA

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industrie 4.0," *WIRTSCHAFTSINFORMATIK*, vol. 56, no. 4, pp. 261–264, Aug. 2014, doi: 10.1007/s11576-014-0424-4.

[2] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015, doi: 10.1016/j.mfglet.2014.12.001.

[3] R. Y. Zhong, C. Xu, C. Chen, and G. Q. Huang, "Big Data Analytics for Physical Internet-based intelligent manufacturing shop floors," *Int. J. Prod. Res.*, vol. 55, no. 9, pp. 2610–2621, May 2017, doi: 10.1080/00207543.2015.1086037.

[4] B. Y. Qi, Q. L. Yang, and Y. Y. Zhou,

"Application of AGV in intelligent logistics system," in *Fifth Asia International Symposium on Mechatronics (AISM 2015)*, 2015, pp. 1–5.

[5] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," *Auton. Robots*, vol. 4, no. 1, pp. 7–27, 1997, doi: 10.1023/A:1008855018923.

[6] G. Dudek, M. R. M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Auton. Robots*, vol. 3, no. 4, pp. 375–397, 1996.

[7] A. Rosenfeld, N. Agmon, O. Maksimov, and S. Kraus, "Intelligent agent supporting human–multi-robot team collaboration," *Artif. Intell.*, vol. 252, pp. 211–231, 2017.

[8] J. Gao, X. Hu, and C. Wu, "Design and simulation of multi-robot logistic system," in *2006 2nd IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications*, 2006, pp. 1–6.

[9] I. Andersone, "The characteristics of the map merging methods: A survey," *Appl. Comput. Syst.*, vol. 41, no. 1, pp. 113–121, 2010.

[10] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2005, pp. 24–29.

[11] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual SLAM: Applications to mobile robotics," *Intell. Ind. Syst.*, vol. 1, no. 4, pp. 289–311, 2015.

[12] J. Shimamura, M. Morimoto, and H. Koike, "Robust vSLAM for Dynamic Scenes.," in *MVA*, 2011, pp. 344–347.

[13] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.

[14] A. Buyval, I. Afanasyev, and E. Magid, "Comparative analysis of ROS-based monocular SLAM methods for indoor navigation," in *Ninth International Conference on Machine Vision (ICMV 2016)*, 2017, vol. 10341, p. 103411K.

[15] P. Anggraeni, M. Mrabet, M. Defoort, and M. Djemai, "Development of a wireless communication platform for multiple-mobile robots using ROS," in *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, 2018, pp. 1–6.

[16] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *J. F. Robot.*, vol. 36, no. 4, pp. 763–781, 2019.

[17] Wikipedia contributors, "Robotino," *Wikipedia, The Free Encyclopedia.*, 2019. https://en.wikipedia.org/w/index.php?title=Robotino&oldid=898008008 (accessed Jan. 12, 2020).

[18] S. F. R. Alves, J. M. Rosario, H. Ferasoli Filho, L. K. Rincon, R. A. Yamasaki, and A. Barrera, "Conceptual bases of robot navigation modeling control and applications," *Adv. Robot Navig.*, p. 26, 2011.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[20] M. Andersson and M. Baerveldt, "Simultaneous localization and mapping for vehicles using ORB-SLAM2." Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2018.

[21] A. Fusiello, "Elements of geometric computer vision," *Available fro m http//homepages. inf. ed. ac. uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO4/tutorial. html*, 2006.

[22] Y. Pyo, H. Cho, J. Ryuwoon, and T. Lim, *Robot Programming From The Basic Concept To Practical Programming and Robot Application*. 2017.

[23] Open Source Robotics Foundation, "Setting up your robot using tf," *http://wiki.ros.org*, 2015. http://wiki.ros.org/navigation/Tutorials/RobotSetup/TF (accessed Jun. 06, 2020).

[24] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[25] C. Luo, W. Yang, P. Huang, and J. Zhou, "Overview of Image Matching Based on ORB Algorithm," in *Journal of Physics: Conference Series*, 2019, vol. 1237, no. 3, p. 32020.

Biodata Penulis

Pipit Anggraeni, lahir di Bandung, 24 Agustus 1979. Merupakan staf pengajar di program studi Teknologi Rekayasa Otomasi, jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung sejak tahun 2000-sekarang. Menyelesaikan pendidikan Diploma 3 di Politeknik Manufaktur Bandung di jurusan Teknik Otomasi Manufaktur dan Mekatronika dengan program studi Mekatronika pada tahun 2000. Sarjana Teknik di Jurusan Teknik Elektro, Universitas Indonesia pada tahun 2004. Tahun 2011 memperoleh gelar Magister Teknik dan Master Science & Technology (Program Double Degree Indonesia Prancis - DDIP) di jurusan Teknik Elektro, Universitas Indonesia dan Universitas d'Angers, ISTIA, Angers, Prancis, dengan bidang konsentrasi Kontrol Industri dan Kontrol & Teknik Informatik. Program Doctoral diselesaikan pada tahun 2019 di Laboratory of Industrial and Human Automation control, Mechanical engineering and Computer Science (LAMIH), Universitas Polytechnic Hauts de France, Valenciennes, Prancis, dengan bidang konsentrasi sistem multi-agent untuk robotik.

Ridwan, lahir di Muara Teweh, 12 Juni 1978. Merupakan staf pengajar di program studi Teknologi Rekayasa Otomasi, jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung sejak tahun 2000-sekarang. Menyelesaikan pendidikan Diploma 3 di Politeknik Manufaktur Bandung di jurusan Teknik Otomasi Manufaktur dan Mekatronika dengan program studi Mekatronika pada tahun 2000. Sarjana Sains Terapan di Jurusan Teknik Elektro kosentrasi Teknologi Informasi, Politeknik Elektronika Negeri Surabaya pada tahun 2004. Tahun 2016 memperoleh gelar Master Engineering di Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada Yogyakarta dengan bidang konsentrasi Isyarat Sistem Elektronis.

Muhammad Taufiq Aulia Asshydiqi, lahir di Cimahi, 23 Juli 1996. Merupakan mahasiswa tingkat akhir jenjang Diploma IV di program studi Teknologi Rekayasa Otomasi, jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung tahun ajaran 2019/2020.