

BAB II

TINJAUAN PUSTAKA

Dalam pengerjaan penelitian ini dapat terbagi menjadi dua tahapan, yaitu pengembangan sistem navigasi di *hexagonal grid workspace* dan arsitektur sistem layanan komputasi untuk mengontrol dan memantau AGV. Oleh karena itu, dalam tinjauan pustaka ini akan membahas dua jenis topik penelitian terdahulu yaitu mengenai sistem navigasi di *hexagonal grid workspace* dan arsitektur sistem.

A. Sistem Navigasi di *Hexagonal Grid Workspace*

Penelitian terkait sistem navigasi di *hexagonal grid workspace* telah pernah dilakukan sebelumnya. Pada *hexagonal grid workspace*, lingkungan kerja dibagi menjadi sel-sel berbentuk heksagon berukuran sama dengan informasi binari. Informasi yang disimpan dapat berupa nilai binari nol atau satu, dimana nol berarti tidak ada rintangan pada sel tersebut dan satu berarti terdapat rintangan pada sel tersebut. Berdasarkan informasi tersebut dilakukan pencarian rute yang bebas rintangan. Pada penelitian terdahulu, beberapa algoritma pencarian rute yang telah dieksplorasi adalah algoritma A* [9] [17] dan algoritma *Ant Colony Optimization* (ACO) [8].

An dkk. [9] mengeksplorasi pengembangan algoritma A* di lingkungan dengan berbagai jenis medan, dimana terdapat sel rumput yang dapat mengganggu pergerakan robot jika dilewati. Untuk mengatasi hal tersebut, An dkk. mengimplementasikan strategi biaya yang fleksibel dan fungsi heuristik adaptif. Dengan demikian, biaya melewati sel rumput akan lebih mahal daripada sel biasa. Berdasarkan hasil pengujian, algoritma A* yang dikembangkan berhasil mencari rute teroptimal dengan biaya minimal di lingkungan dengan berbagai jenis medan.

Wang dkk. [17] juga mengeksplorasi pengembangan A* di lingkungan dengan berbagai jenis medan, dimana terdapat *landmark* dan pejalan kaki. Wang dkk. menerapkan algoritma *Kernel Density Estimation* yang mengestimasi distribusi posisi *landmark* dan pejalan kaki yang menghasilkan peta distribusi objek dalam bentuk *hexagonal grid*. Fungsi

heuristik A* dimodifikasi untuk memperhitungkan tiga faktor, yaitu jarak, rekognisi, dan kepadatan objek sekitar. Berdasarkan hasil pengujian, algoritma A* yang dikembangkan efektif memperhitungkan faktor-faktor tersebut dalam mencari rute, tetapi masih membutuhkan pengembangan dalam keefisiensannya.

Berbeda dengan penelitian An dkk. dan Wang dkk., Shao dkk. [8] mengeksplorasi pengembangan algoritma ACO di *hexagonal grid workspace*. Shao dkk. menerapkan beberapa pengembangan pada algoritma ACO, yaitu distribusi gauss untuk pendistribusian feromon yang tidak merata pada tahap awal, peningkatan probabilitas transisi untuk menghindari *deadlock*, fungsi heuristik adaptif, dan pembaruan aturan feromon untuk pencarian yang lebih efektif dan efisien. Berdasarkan hasil pengujian, algoritma ACO yang dikembangkan menghasilkan rute yang lebih pendek dan jumlah belokan lebih kecil daripada algoritma ACO tradisional.

B. Penerapan Jaringan Komunikasi Pada AGV

Dalam penerapannya, jaringan komunikasi pada AGV dimanfaatkan untuk memantau data atau mengontrol AGV. Untuk pemanfaatan ini dapat menggunakan koneksi *bluetooth* atau WIFI untuk menghubungkan aplikasi monitor dan kontrol dengan AGV [16] [18]. Selain untuk memantau dan mengontrol, jaringan komunikasi juga dapat dimanfaatkan untuk memindahkan komputasi yang berat pada AGV ke perangkat komputasi eksternal dengan komunikasi HTTP antara server dan AGV [19] atau menerapkan arsitektur *edge-fog-cloud* [20] [21] [22].

Balakrishna dan Rajesh [18], dan Loh dan Muhammed [16] memanfaatkan jaringan komunikasi untuk memantau dan mengontrol AGV. Balakrishna dan Rajesh mengembangkan AGV untuk pemotong rumput yang dapat dikontrol melalui aplikasi android dengan koneksi *bluetooth* melalui tujuh pesan kontrol. Sedangkan, Loh dan Muhammed mengembangkan AGV pengangkut barang yang dapat dipantau dan

dikontrol melalui aplikasi web dengan koneksi WIFI. AGV bergerak mengikuti garis di lantai dan akan berhenti ketika ada rintangan di depannya atau perintah dari aplikasi web.

Pemanfaatan jaringan komunikasi untuk perangkat komputasi eksternal dengan komunikasi HTTP antara server dan AGV dilakukan oleh Junior dkk. [19]. Junior dkk. mengembangkan aplikasi web yang melakukan klasifikasi objek menggunakan *convolution neural network* (CNN) untuk ekstraksi fitur dan *machine learning* untuk klasifikasi. Berdasarkan klasifikasi tersebut dilakukan lokalisasi posisi AGV sesuai data peta yang telah tersimpan.

Penerapan arsitektur *edge-fog-cloud* untuk pembagian komputasi dilakukan oleh Boshoff dkk. [21], Sarker dkk. [22], Bhalekar dan Penamakuri [20] dengan fungsi yang berbeda-beda. Pada tingkat *edge*, yaitu robot, hanya dilakukan pembacaan sensor dan pemrosesan sederhana. Dilanjutkan di tingkat *fog*, dilakukan pengambilan keputusan yang dibutuhkan segera oleh robot dan penyimpanan data sementara. Pada tingkat *cloud*, dilakukan penyimpanan data permanen dan pengolahan data yang lebih besar.

Boshoff dkk., Bhalekar dan Penamakuri menggunakan tingkat *fog* untuk pengelolaan gambar dari kamera di tingkat *edge*. Boshoff dkk. melakukan pendeteksian penanda dan pose robot lengan. Sedangkan Bhalekar dan Penamakuri melakukan pengenalan objek berdasarkan warna dan pelacakan objek menggunakan algoritma Kalman Filter. Pada kedua penelitian tersebut, ditunjukkan penggunaan arsitektur ini meningkatkan kecepatan pemrosesan dan memungkinkan penggunaan *graphical processing unit* (GPU) pada robot dengan kemampuan komputasi rendah.

Sarker dkk. menguji coba arsitektur *edge-fog-cloud* pada AGV yang dilengkapi sensor *Light Detection and Ranging* (LiDAR) dan terkoneksi dengan komputer melalui koneksi *bluetooth*. Berdasarkan data LiDAR, pada tingkat *fog*, dijalankan algoritma BreezySLAM untuk memetakan lingkungan dan menentukan posisi AGV. Pada penelitian Sarker dkk.

menunjukkan penggunaan arsitektur ini menurunkan pemakaian daya sebanyak 30% oleh LiDAR.

Bhat dkk. [14] juga mengimplementasi arsitektur yang hampir serupa dengan arsitektur *edge-fog-cloud*. Berbeda dengan arsitektur *edge-fog-cloud*, Bhat dkk. menerapkannya di jaringan *wireless local area network* (WLAN). Data dari AGV diteruskan ke komputer pusat dalam satu jaringan untuk melakukan komputasi *Visual SLAM* (VSLAM), perencanaan jalur dengan algoritma A*, dan rekognisi palet menggunakan YOLO. Arsitektur yang dikembangkan diuji coba pada forklif untuk melakukan navigasi menuju titik tujuan dan mengangkat palet di lingkungan gudang yang diketahui sepenuhnya.

Berikut ini Tabel 2.1 merupakan tabel perbandingan penelitian ini dengan penelitian terdahulu:

Tabel 2.1 Tabel perbandingan penelitian terdahulu

Pembanding	Junior dkk. [19]	Sarker dkk. [22]	Bhalekar dan Penamakuri [20]	Bhat dkk. [14]	Ricky Junaidi
Komputasi yang dipindahkan	Klasifikasi objek	Pengenalan objek berdasarkan warna dan pelacakan posisi objek	Deteksi penanda dan pose robot lengan	Lokalisasi, perencanaan jalur, dan rekognisi palet	Perencanaan jalur
Metode Pembagian Komputasi	Web API	<i>edge-fog-cloud</i>	<i>edge-fog-cloud</i>	<i>edge-Center PC</i>	<i>edge-server</i>
Pencarian rute	-	-	-	A*	A*
Lokalisasi	CNN	-	-	VSLAM	BreezySLAM
Model Representasi Lingkungan	Grafik	-	-	<i>Grid Persegi</i>	<i>Hexagonal Grid</i>
Metode Komunikasi	WIFI, HTTP	WIFI	WIFI	WIFI	WIFI, <i>Websocket</i>
Prosesor pada robot	Tidak disebutkan	MCU	Tidak disebutkan	Tidak disebutkan	Arduino Uno dan Raspberry Pi
Sensor pada robot	Kamera	LIDAR	Kamera	Kamera, LIDAR	LIDAR