BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Pencarian jalur terpendek merupakan salah satu masalah yang sampai saat ini masih terus dikembangkan serta dipilih sebagai topik penelitian. Seperti penelitian yang dilakukan oleh (Goldberg, 2007) yang membandingkan beberapa algoritma pencarian jalur terpendek. Algoritma yang dibahas adalah algoritma Djikstra, algoritma Djikstra dengan dua arah, algoritma A*, serta beberapa algoritma yang menggunakan data-data yang sudah diolah sebelumnya untuk digunakan pada algoritma pencarian jalur selanjutnya.

Memodifikasi sebuah algoritma yang sudah ada juga merupakan salah satu alternatif yang dipilih oleh para peneliti sebelumnya untuk melakukan optimisasi algoritma pencarian jalur terpendek. Penelitian (Yakovlev, et al., 2015) menjelaskan tentang modifikasi dari algoritma A* pada *grid-based environment*. Yakovlev, et al. memperkenalkan algoritma LIAN yang merupakan modifikasi dari algoritma A* yang berguna untuk menemukan jalur yang baik (*smooth*) daripada jalur yang terpendek. Penelitian (Qu, et al., 2015) juga menjelaskan tentang algoritma *Spiking Neural Network* (SNN) yang mampu menjawab kekurangan dari algoritma Djikstra yang dinilai sangat membebani kerja dari CPU.

Kemampuan dari A* dalam menemukan jalur terpendek, membuat penggunaan algoritma A* terutama pada dunia game terus meningkat. Seperti penelitian yang dilakukan oleh (Lee & Lawrence 2013) atau yang dilakukan (Lawrence & Bulitko

2013), dimana algoritma A* digunakan pada game Dragon Games™ atau *Counter-Strike: Source*.

Penelitian (Lee & Lawrence, 2013) berusaha memecahkan masalah besarnya jumlah node yang perlu dikunjungi dan banyaknya jumlah agen yang perlu dikenakan algoritma pencarian jalur terpendek. Lee dan Lawrence menjawabnya dengan membuat sebuah database dari node-node yang sudah biasa dilewati. Proses pencarian jalur terpendek akan diawali dengan mencari node-node yang sudah ada pada database terlebih dahulu, sebelum akhirnya dilakukan pencarian menggunakan algoritma A*. Hal ini akan menghemat penggunaan memori serta mempercepat proses pencarian jalur terpendek.

Penelitian (Lawrence & Bulitko, 2013) selanjutnya menjawab kekurangan dari pencarian jalur terpendek yang menggunakan database sebelumnya. Salah satu kelemahannya adalah penggunaan database pada pencarian jalur terpendek memungkin besarnya waktu komputasi awal serta memungkinkan tidak ditemukannya semua kemungkinan jalur yang dapat dilalui. Mereka mengatasinya dengan membagi proses pencarian dari node awal ke node akhir menjadi bagian-bagian yang lebih kecil. Hal ini akan mempercepat proses awal komputasi serta ruang pencarian akan dilalui semua.

Penelitian lebih lanjut menyebutkan bahwa dalam kasus-kasus tertentu pencarian jalur terpendek perlu dilakukan secara *real-time* karena terus berubahnya lingkungan di sekitar NPC. Penelitian yang sudah dilakukan (Cowling, et al., 2014)

menyebutkan bahwa pencarian jalur terpendek dapat dilakukan secara *real-time*, yang sedikit berbeda dari pencarian jalur terpendek konvensional pada umumnya.

Proses paralel juga dipandang sebagai salah satu teknik untuk optimisasi algoritma-algoritma pencarian jalur terpendek. Penelitian (Brand & Bidarra, 2011) menerapkan algoritma *Parallel Ripple Search* (PRS) untuk menyelesaikan banyaknya agen yang perlu bergerak pada waktu yang bersamaan. Penelitian yang dilakukan oleh (Rios & Chaimowicz, 2011) juga menggunakan proses paralel dimana mereka membuat algoritma *Parallel New Bidirectional A** (PNBA*) untuk mempercepat proses pencarian jalur terpendek. Proses pencarian jalur dapat berjalan lebih cepat karena proses pencarian menggunakan dua cores CPU secara bersamaan.

Proses pencarian jalur terpendek menggunakan GPU juga sudah dapat ditemukan. Pada penelitian yang dilakukan oleh (Inam, 2010) dimana melalui penelitiannya yang menggunakan CUDA, proses pencarian jalur terpendek menggunakan algoritma A* dapat berjalan hingga 2000 agen yang berjalan secara bersamaan. Hasilnya, proses pencarian jalur terpendek satu agen menggunakan CPU hampir sama dengan 2000 agen menggunakan GPU.

Berbagai cara telah dilakukan oleh para peneliti sebelumnya untuk melakukan optimisasi pada algoritma-algoritma jalur terpendek, khususnya A*. Penelitian-penelitian itu dirangkum dan disajikan pada Tabel 1.

Tabel 1 Perbandingan penelitian terdahulu

Peniliti	**Masalah**	**Solusi**	**Keterbatasan**
Brand & Bidarra (2011)	Banyaknya agen yang bergerak secara bersamaan pada lingkungan yang besar.	Dibuatnya algoritma *Parallel Ripple Search* (PRS).	PRS belum mampu menentukan dengan tepat *cores* mana yang perlu mengerjakan pencarian jalur. PRS masih dapat dikembangkan dengan optimasi memori yang digunakan *cores* secara bersamaan.
Yakovlev, et al. (2015)	Menemukan jalur terbaik dengan sudut berbelok terbaik.	Dibuatnya algoritma LIAN, yang merupakan modifikasi dari algoritma A*.	LIAN bisa disempurnakan lagi dengan merubah nilai Δ.
R. Anbuselvi (2013)	Menemukan jalur terpendek untuk *grid-base environment*.	Penggunaan algoritma A* pada pencarian pohon BSP.	
Lee & Lawrence (2013)	Optimisasi algoritma A* agar A* menggunakan memori yang kecil dan memiliki *response time* lebih cepat.	Menerapkan algoritma Database A* (DBA*).	Masih belum mampu menangani perubahan-perubahan pada *grid*.
Lawrence & Bulitko (2013)	Penggunaan database pada pencarian jalur terpendek akan menambah waktu pencarian awal serta memungkinkan tidak semua jalur akan ditemui.	Menerapkan abstraksi jalur, membagi proses pencarian jalur menjadi proses-proses yang lebih kecil.	Algortima yang dipaparkan belum mampu menangani proses jalur pencarian pada peta yang besar.

Qu, et al. (2015)	Algoritma Djikstra memberikan beban kepada CPU terlalu tinggi, terutama ketika melakukan perhitungan jalur terpendek yang memiliki node-node yang kompleks.	Penggunaan Spiking Neural Network (SNNs) dengan model *integrate-and-fire* untuk menyelesaikan masalah jalur terpendek.	Algoritma SNNs hanya dibandingkan dengan Djikstra.
Rios & Chaimowicz (2011)	Optimisasi algoritma A* pada *Grid-Based Environment*	Digunakannya *Parallel New Bidirectional* A* (PNBA*)	PNBA* masih terbatas karena hanya menggunakan 2 cores dalam perhitungan
Peneliti	Optimisasi Algoritma A* pada *Hexagon-Based Environment*	Menerapkan *Parallel Bidirectional Search* pada Algoritma A*.	

B. Model Penelitian

Model yang digunakan pada penelitian ini adalah pengamatan dan pengembangan dari penelitian sebelumnya. Pada penelitian yang dilakukan oleh (Rios & Chaimowicz, 2011), mereka mampu menerapkan PNBA* dalam *Grid-Based Environment*. Penelitian ini akan menerapkan PBS pada algoritma A* di *Hexagon-Based Environment*.