

1. Pendahuluan

1.1 Latar Belakang Masalah

Masalah pencarian jalur terpendek (*shortest path*) dalam graf merupakan masalah yang sudah cukup banyak dibahas. Dimana masalah tersebut merupakan masalah optimasi[1] yaitu suatu masalah yang memungkinkan *cost* terendah dalam penyelesaiannya. Penyelesaian masalah *shortest path* tersebut bisa menggunakan berbagai macam algoritma. Salah satu algoritma yang digunakan adalah algoritma MBDA* (*Modified Bi-directional A**) yang merupakan variasi dari algoritma A*.

Dibandingkan dengan variasi algoritma A* yang lainnya, algoritma MBDA* ini mampu menyelesaikan masalah *shortest path* yang memiliki jumlah *node* yang besar dengan waktu yang lebih cepat[8]. Eksperimen yang dilakukan oleh Tetsuo Shibuya terhadap jaringan jalan sesungguhnya (*actual road network*) di Tokyo membuktikan bahwa jumlah simpul yang dibangkitkan oleh MBDA* adalah setengah dari jumlah simpul yang dibangkitkan oleh A*[9]. Serta algoritma MBDA* ini juga merupakan algoritma yang complete dan optimal[9]. Sesuai dengan namanya maka pencarian *shortest path* dilakukan secara dua arah, yaitu dengan cara membangkitkan node dari *initial node* dan juga membangkitkan node dari *goal node*, sehingga pencarian tersebut berhenti jika node sudah bertemu pada node yang sudah pernah terpilih sebagai *best node*.

Pada tugas akhir ini diusulkan sebuah metode pencarian jalur terpendek dengan cara membagi *path* menjadi dua bagian dengan cara menemukan sebuah titik random yang berada di antara *initial node* dan *goal node* yang kemudian disebut sebagai *middle node*. Kemudian diberlakukan algoritma MBDA* untuk menemukan jalur terpendeknya. Sehingga pencarian jalur terpendek tersebut akan diawali dari mencari jalur dari *initial node* ke *middle node* dengan cara MBDA* lalu dilanjutkan dengan mencari jalur dari *middle node* ke *goal node* dengan cara MBDA* juga. Metode pencarian tersebut selanjutnya bisa disebut dengan Dgraph-MBDA*. Pembagian node ini bertujuan untuk mempersempit ruang pencarian solusi, sehingga proses pencarian solusi menjadi semakin cepat. Disini graf hanya dibagi ke dalam dua bagian, dikarenakan jika terlalu banyak dilakukan pembagian akan mengakibatkan kompleksitas algoritma Dgraph-MBDA* menjadi lebih tinggi yang disebabkan oleh perhitungan untuk menentukan *middle node* yang terlalu banyak, sehingga akan membuat penyelesaian masalah akan menjadi lebih lama.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang akan dibahas adalah :

1. Bagaimana cara menentukan sebuah *node* untuk dijadikan sebagai *middle node*.
2. Bagaimana kecepatan dari algoritma Dgraph-MBDA* jika digunakan untuk mencari jalur terpendek.

3. Apakah solusi yang dihasilkan dari algoritma Dgraph-MBDA* tersebut tetap optimal.
4. Berapa banyak *node* yang dibangkitkan dalam pencarian *shortest path* dengan menggunakan Dgraph-MBDA*.

1.3 Tujuan

Tujuan dari penyusunan tugas akhir ini adalah sebagai berikut :

1. Mengetahui bagaimana cara menentukan sebuah *node* untuk dijadikan sebagai *middle node*.
2. Membuktikan apakah dengan algoritma Dgraph-MBDA* lebih cepat dalam menemukan solusi *shortest path*.
3. Membuktikan apakah dengan menggunakan algoritma Dgraph-MBDA* tersebut akan dihasilkan solusi yang tetap optimal, sesuai dengan karakteristik MBDA*.
4. Mengetahui apakah jumlah *node* yang dibangkitkan oleh Dgraph-MBDA* lebih sedikit dibandingkan dengan algoritma MBDA*

1.4 Batasan Masalah

Beberapa batasan masalah dalam pengerjaan tugas akhir ini adalah :

1. Studi kasus yang digunakan adalah peta Kota Madya Bandung,
2. Masalah yang dibahas adalah mengenai performa dari algoritma Dgraph-MBDA* dalam menemukan solusi.
3. Algoritma Dgraph-MBDA* akan dibandingkan dengan algoritma MBDA*.
4. Bobot dihitung berdasarkan jarak.

1.5 Metodologi Penyelesaian Masalah

1. Studi Literatur
Pencarian referensi yang berhubungan dengan algoritma MBDA*, dan *shortest path*.
2. Pencarian Data
Pencarian data yang akan dijadikan sebagai sampel untuk menerapkan algoritma Dgraph-MBDA*.
3. Analisa dan Desain
Merupakan tahapan yang meliputi analisa dan desain dari penggunaan algoritma Dgraph-MBDA* dalam menyelesaikan masalah *shortest path*.
4. Implementasi
Merupakan tahapan mengimplementasikan kinerja dari algoritma Dgraph-MBDA* dalam menyelesaikan masalah *shortest path*.
5. Analisa hasil
Melakukan analisa dan perbandingan kerja dari algoritma MBDA* dan Dgraph-MBDA* pada masalah penyelesaian *shortest path*.

6. Pengambilan keputusan dan penyusunan laporan tugas akhir

Pengambilan keputusan dari hasil analisa yang telah dilakukan pada tahap sebelumnya dan kemudian dilakukan penyusunan laporan terhadap analisa yang dilakukan.