



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB I

PENDAHULUAN

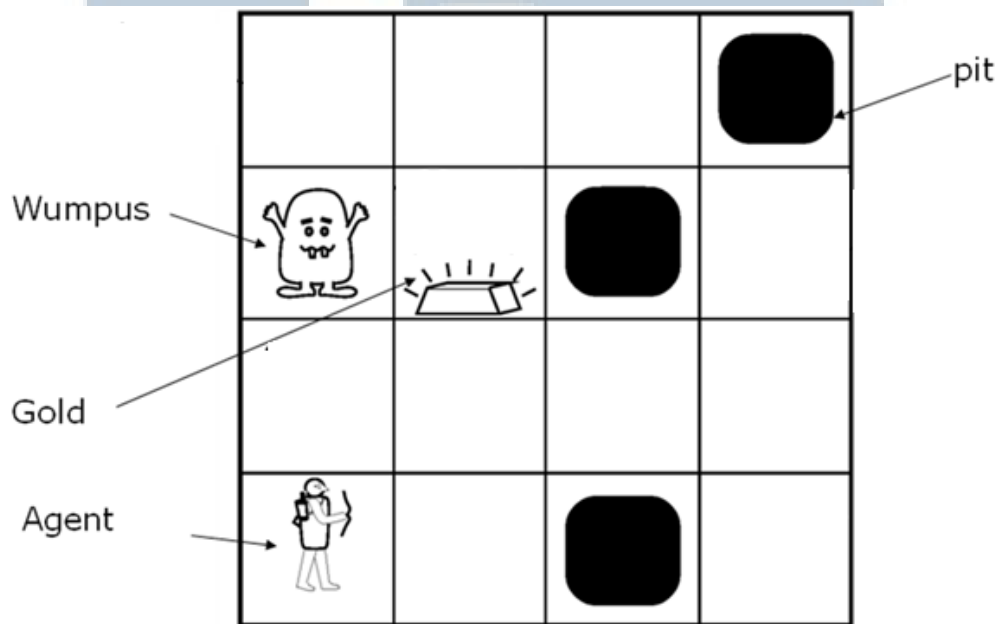
1.1 Latar Belakang Permasalahan

Di tengah masyarakat dengan aktivitas yang tinggi, mobilitas menjadi hal yang penting. Namun pada kenyataannya, terdapat banyak hal yang dapat menghambat mobilitas, seperti kemacetan, perbaikan jalan, maupun banjir. Hambatan-hambatan memaksa pengguna jalan menentukan pilihan untuk mengambil rute alternatif atau tetap melalui hambatan tersebut. Tanpa adanya informasi yang jelas, keputusan diambil berdasarkan naluri semata sehingga pilihan dapat merugikan.

Kondisi tersebut serupa dengan permainan “*Hunt The Wumpus*”. Pada permainan ini, pemain (*agent*) harus mencapai kotak berisi emas (*gold*) sembari menghindari lubang (*pit*) dan monster (*wumpus*). Pemain tidak dapat melewati lubang namun dapat melewati wumpus, di mana pemain harus membunuh wumpus terlebih dahulu (lihat Gambar 1.1). Pada dunia nyata, lubang melambangkan hambatan yang tidak dapat dilalui seperti jalan yang ditutup, dan wumpus melambangkan hambatan yang dapat dilalui seperti kemacetan. Setiap hambatan tentu akan mempengaruhi waktu yang diperlukan untuk mencapai tujuan, yang dalam permainan disebut dengan *cost*.

Lokasi awal pada permainan akan ditentukan dan dapat diubah oleh pemain, sedangkan lokasi akhir tetap dan tak dapat diubah. Konsep ini dapat diterapkan untuk sistem *delivery* pada restoran dengan banyak cabang, di mana restoran

dapat menentukan cabang yang terdekat dengan lokasi pengiriman. Selain itu, konsep ini dapat digunakan pada sistem *Global Positioning System Online (GPS Online)*, di mana *GPS* dapat meng-*update* jalur berdasarkan lokasi pemakai atau perubahan parameter seperti tingkat kemacetan.



Gambar 1.1 Gambaran Permainan “*Hunt The Wumpus*”

Permainan “*Hunt The Wumpus*” dapat dipetakan menjadi *graph* yang dapat dikategorikan sebagai *sparse graph*. *Sparse graph* merupakan sebuah *graph* yang memiliki jumlah *edge* yang tidak terlalu banyak di mana $|E| < |V|^2$ (Cormen, dkk., 2001: 527).

Pada “*Hunt The Wumpus*”, semakin kecil *cost* yang dibutuhkan untuk mencapai kotak berisi emas, maka akan semakin baik. Untuk itu, pemain harus mempertimbangkan setiap jalur yang ada. “*Hunt The Wumpus*” dilengkapi dengan kecerdasan buatan di mana komputer dapat menentukan rute yang tepat

untuk mencapai tujuan. Untuk memilih rute dengan *cost* yang minimal, maka dibutuhkan algoritma pencarian untuk menentukan jalur terpendek (*shortest path algorithm*).

Secara umum, terdapat lima *shortest path algorithm* yang sering digunakan, yaitu Dijkstra, A* Search, Bellman-Ford, Johnson, dan Floyd-Warshall. Setiap algoritma memiliki karakteristik dan kelebihan yang berbeda-beda. Hal inilah yang menyebabkan tidak semua algoritma dapat diterapkan dalam permainan "Hunt The Wumpus". Pada permainan ini, hanya terdapat satu tujuan yaitu kotak berisi emas, namun pemain dapat menentukan titik awal sehingga titik awal dapat terletak di mana saja kecuali pada kotak berisi lubang. Setiap perpindahan dari satu kotak ke kotak lain membutuhkan *cost*, di mana *cost* selalu bernilai positif.

Algoritma Dijkstra merupakan sebuah *graph search algorithm* yang menyelesaikan *single-source shortest path problem* di mana Dijkstra akan mencari jalur terpendek dari satu *start vertex* dengan cara memeriksa dan membandingkan setiap jalur. Walaupun demikian, Dijkstra dapat dimodifikasi sehingga dapat digunakan untuk mencari jalur terpendek dari setiap *vertex*. Untuk *sparse graph*, yaitu *graph* dengan jumlah edge yang lebih kecil dari V^2 , Dijkstra dapat memiliki *time complexity* yang lebih kecil (Cormen, dkk., 2001: 599). Sama seperti Dijkstra, Bellman-Ford menghitung *single-source shortest path* dan memiliki struktur yang mirip dengan Dijkstra, namun lebih lambat. Kelebihan dari algoritma ini adalah dapat digunakan pada *graph* dengan *negative edge weights*, yang tidak ditemukan pada "Hunt The Wumpus". Dengan demikian, Dijkstra akan

lebih cocok diterapkan dalam permainan dibandingkan dengan Bellman-Ford. (Cormen, dkk., 2001: 588)

Algoritma Dijkstra mengalami banyak pengembangan dan salah satunya adalah algoritma A* Search. A* Search memiliki performa yang lebih tinggi, di mana algoritma ini menggunakan *heuristic function*. *Heuristic* adalah studi mengenai metode dan aturan dalam penemuan solusi. Contoh *heuristic function* dalam *route finding problem* adalah panjang garis lurus dari suatu *vertex* ke *vertex* tujuan, karena garis lurus merupakan jalur tercepat yang dapat dilalui (Russel, 2010: 92). Algoritma Dijkstra dapat dilihat sebagai *special case* dari A* di mana fungsi *heuristic* $h(x) = 0$ untuk setiap x , sehingga untuk kasus yang tidak menggunakan *heuristic* seperti "Hunt The Wumpus", lebih baik menggunakan Dijkstra daripada A*.

Algoritma lain yang sering digunakan adalah Johnson dan Floyd-Warshall. Berbeda dengan Dijkstra yang mencari jalur terpendek dari suatu *start node*, algoritma Johnson dan Floyd-Warshall akan mencari jalur terpendek antara setiap *node* yang berpasangan. Kedua algoritma ini memiliki *time complexity* yang sama yaitu $O(n^3)$ untuk $n =$ jumlah *node*. Walaupun demikian, kedua algoritma tersebut memiliki karakteristik yang bertolak belakang di mana Floyd-Warshall akan bekerja optimal pada *dense graph* yaitu *graph* dengan jumlah *edge* mendekati jumlah maksimal dari *edge* (Cormen, dkk., 2001: 629) dan Johnson bekerja optimal pada *sparse graph* yaitu *graph* dengan jumlah *edge* yang sedikit (Cormen, dkk., 2001: 636). Algoritma Johnson merupakan penggabungan dari Bellman-Ford untuk menghilangkan *negative weight* dan Dijkstra untuk mencari

jalur. Permainan “*Hunt The Wumpus*” tidak memiliki *negative weight*, sehingga pemakaian Johnson tidak tepat.

Berdasarkan analisis di atas, maka *shortest path algorithm* yang sesuai dengan karakteristik “*Hunt The Wumpus*” adalah Dijkstra dan Floyd-Warshall. Secara teori, Dijkstra yang sudah dimodifikasi memiliki *time complexity* yang lebih besar dibandingkan dengan Floyd-Warshall, namun Dijkstra bekerja optimal pada *sparse graph* sehingga akan bekerja lebih optimal pada permainan “*Hunt The Wumpus*”. Berdasarkan hal tersebut, penulis memutuskan untuk membuat skripsi dengan judul “Perbandingan Algoritma Floyd-Warshall dan Dijkstra pada Permainan *Hunt The Wumpus*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, penulis akan mengajukan rumusan masalah yaitu sebagai berikut.

1. Bagaimanakah cara menerapkan algoritma Floyd-Warshall dan Dijkstra pada permainan “*Hunt The Wumpus*”?
2. Bagaimanakah hasil perbandingan performa dari algoritma Floyd-Warshall dan Dijkstra pada permainan “*Hunt The Wumpus*”?

1.3 Batasan Masalah

Analisis perbandingan algoritma Dijkstra dan Floyd-Warshall dilakukan berdasarkan data yang didapat dari permainan “*Hunt The Wumpus*”.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan penelitian ini:

1. Menerapkan algoritma Floyd-Warshall dan Dijkstra pada permainan “Hunt The Wumpus”.
2. Mendapatkan hasil perbandingan performa yaitu *completeness*, *optimality*, *space complexity*, and *time complexity* dari algoritma Floyd-Warshall dan Dijkstra pada permainan “Hunt The Wumpus”.

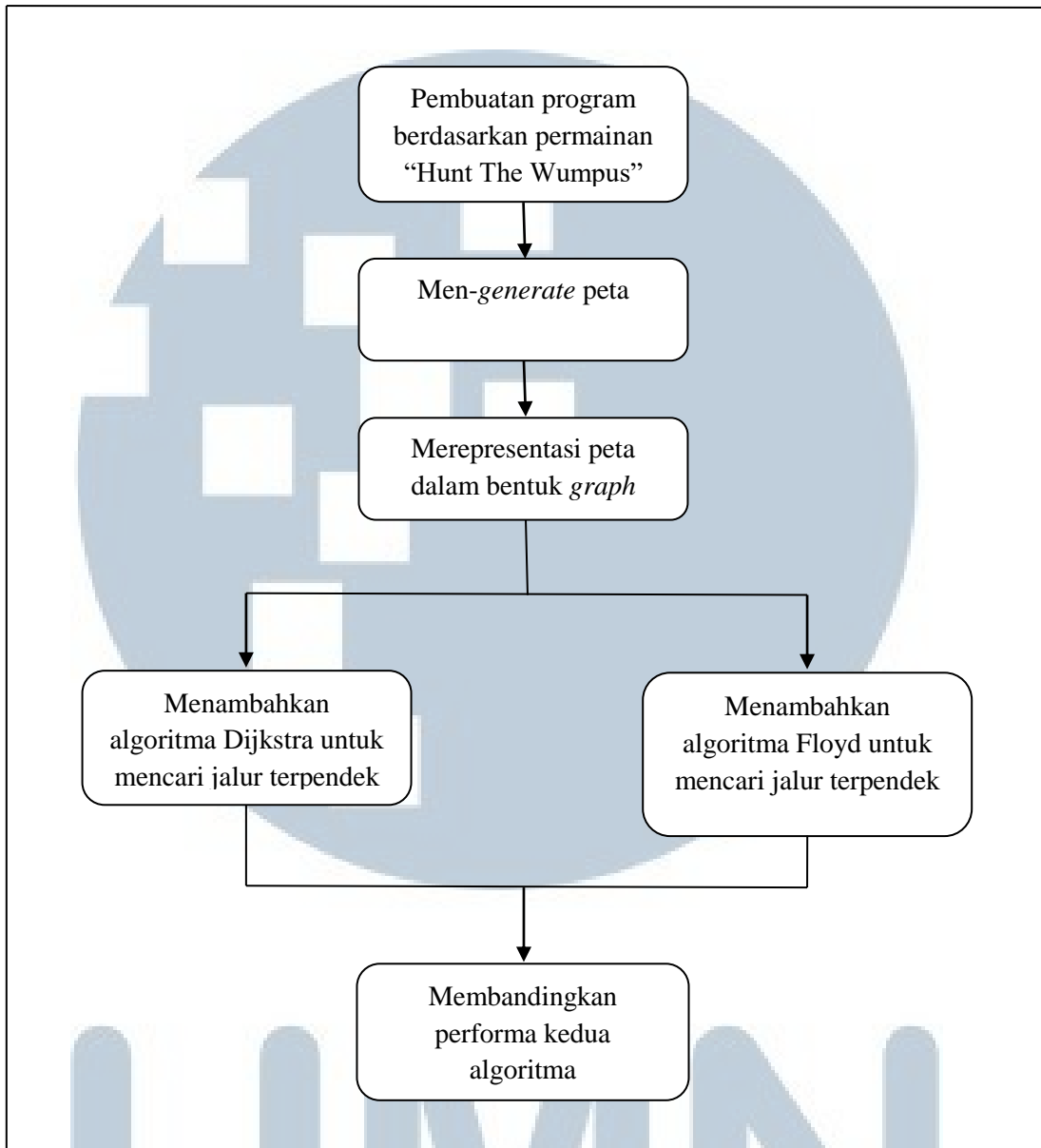
1.5 Manfaat Penelitian

Manfaat dari hasil penelitian ini digunakan untuk meningkatkan kemampuan teknologi untuk GPS saat ini agar bisa memberikan fitur *shortest path* dan *fastest path* (jika diintegrasikan dengan data halangan saat dijalankan).

1.6 Metodologi Penelitian

Penelitian akan dilakukan dengan tahapan pembuatan program berdasarkan permainan “Hunt The Wumpus”, men-*generate* peta, memetakan permainan dalam bentuk *graph*, menambahkan algoritma Dijkstra dan Floyd-Warshall, dan dilanjutkan dengan membandingkan performa kedua algoritma dengan diagram kerja seperti terlihat pada gambar 1.2.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 1.2 Diagram Kerja

U I V N
UNIVERSITAS
MULTIMEDIA
NUSANTARA

1.7 Sistematika Penulisan

Jarak antara baris satu dengan yang lain dibuat dua spasi. Sedangkan abstrak dan daftar pustaka menggunakan satu spasi.

BAB I. PENDAHULUAN

berisikan latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian yang dilakukan, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II. TELAAH LITERATUR

dalam bab ini dibahas mengenai algoritma terutama Dijkstra dan Floyd-Warshall, *graph*, permainan “*Hunt The Wumpus*”, struktur data, pemrograman berorientasi obyek, dan perbandingan algoritma.

BAB III. METODE PENELITIAN

berisikan uraian metode yang digunakan dalam pembuatan program dan analisis perbandingan yang dilakukan.

BAB IV. HASIL DAN PEMBAHASAN

berisikan analisis dan pembahasan dari perbandingan yang didapat dari algoritma Dijkstra dan Floyd-Warshall.

BAB V. SIMPULAN DAN SARAN

berisikan kesimpulan hasil analisis dan pembahasan dari perbandingan yang didapat dari algoritma Dijkstra dan Floyd-Warshall.