

Penggunaan Algoritma A* untuk Sistem Navigasi dalam Permainan Komputer

Kevin Jonathan Koswara 13515052
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515052@stei.itb.ac.id

Abstract—Seiring dengan perkembangan ilmu informatika, permainan komputer juga ikut berkembang pesat. Salah satu permainan yang paling diminati adalah permainan bertipe *open world*. Makalah ini berisi tentang bagaimana graf diaplikasikan dalam peta permainan komputer, algoritma A* untuk mencari jalan terpendek dalam graf, serta pengaplikasiannya dalam sistem navigasi untuk permainan komputer bertipe *open world*.

Keywords—Graf, *Open world*, Dijkstra, Greedy Best-First Search, *waypoint graph*.

I. PENDAHULUAN

Dalam beberapa tahun terakhir, permainan komputer berkembang sangat pesat, mulai dari grafis, sistem, bahkan sekarang banyak permainan komputer dalam 3 dimensi. Perkembangan ini membuat pemain semakin tertarik dan ketagihan dengan permainan tersebut. Permainan yang membuat ketagihan tentu tidak baik terhadap pemainnya. Mereka jadi menghabiskan hampir seluruh waktunya untuk bermain dan tidak melakukan hal produktif.

Namun, kita dapat memanfaatkan ketagihan tersebut untuk sesuatu yang bermanfaat dengan mengembangkan permainan yang mendidik. Permainan mendidik adalah permainan yang dikembangkan untuk melatih kemampuan tertentu dari pemainnya, baik kemampuan aritmatika, bahasa, kreativitas, atau bahkan kemampuan berpikir yang kritis dan strategis. Semakin menarik permainannya, maka pemainnya akan semakin tertarik untuk mempelajarinya lebih dalam.

Salah satu tipe permainan yang banyak menarik minat adalah tipe permainan *open world*. Permainan *open world* adalah permainan di mana pemainnya dapat menjelajahi dunia virtual dalam permainan sesuka hatinya. Pemain bebas menjelajahi dunia ke manapun dan kapanpun dia mau. Permainan bertipe *open world* pertama kali adalah *Ultima* yang dirilis pada tahun 1984.[1] Salah satu permainan *open world* yang terkenal adalah *Grand Theft Auto*, *Minecraft*, dan *Harvest Moon*. *Grand Theft Auto V* merupakan salah satu permainan *open world* dengan dunia yang sangat luas dan banyak dimainkan orang.

Namun, pada dunia yang luas, mustahil bagi seorang pemain baru untuk mengetahui lokasi suatu bangunan

spesifik dalam dunia tersebut. Untuk itu, diperlukan sebuah sistem navigasi dalam permainan *open world* agar seluruh pemain, baik yang lama atau baru, dapat menikmati permainan tersebut.



Gambar 1. Screenshot dari permainan *Grand Theft Auto V*. Sumber:

http://www.gamersyde.com/news_our_videos_of_gta_v-14619_en.html 7 Desember 2016

II. OPEN WORLD

Permainan komputer saat ini sudah banyak berkembang dan dikelompokkan ke dalam beberapa tipe. Tipe permainan komputer sangat banyak, mulai dari *action*, *adventure*, *role playing (RPG)*, *simulation*, *strategy*, dan lain-lain. Setiap tipe tersebut juga dijabarkan menjadi beberapa tipe lain, misalnya tipe RPG dapat dijabarkan antara lain *First Person Shooter (FPS)*, *Action RPG*, *Massive Multiplayer Online RPG (MMORPG)*, *Sandbox*, dan *Open World*.

Permainan bertipe *Role Playing* atau RPG berarti pemain mengambil peran sebagai sebuah karakter di dalam permainan tersebut. Salah satu permainan RPG yang terkenal adalah *Pokemon*, di mana pemain berperan sebagai seorang anak yang bertualang di dunia yang berisi *pokemon*, menangkap dan melatihnya, mengalahkan *gym* dan *champion*.

Permainan bertipe *Open World* memiliki karakteristik *free roaming*, yang berarti pemain dapat bebas menjelajahi dunia dan dapat menjalankan permainan secara non-linier.

Non-linier berarti suatu kejadian tidak harus terjadi pada urutan dan waktu yang sama setiap kali pemain mengulangi permainan tersebut.[1]

Selain *free roaming*, terdapat satu tipe permainan *open world* yang disebut *sandbox*. Permainan bertipe ini memberikan kebebasan pada pemain untuk melakukan apa saja, membangun dunia sesuka hati, atau menjadi siapa saja. Permainan bertipe *sandbox* kebanyakan juga bersifat *open world*. Contoh permainan ini adalah *Minecraft*. Dalam *Minecraft*, pemain sama sekali tidak diberi tujuan atau jalan cerita. Pemain berperan sebagai Steve yang dapat menjelajahi dunia *Minecraft*. Dalam permainan ini, pemain dapat melakukan banyak hal, antara lain membangun rumah, berburu hewan, mengalahkan musuh, bertahan hidup, membuat peralatan, dan lain-lain. Namun, tidak ada kata “tamat” dalam permainan ini karena sama sekali tidak memiliki jalan cerita.[1]

III. GRAF

A. Definisi Graf

Sebuah Graf didefinisikan sebagai kumpulan simpul dan sisi $G = (V, E)$ dengan V adalah himpunan simpul dan E adalah himpunan sisi. Sisi adalah sebuah garis yang menghubungkan 2 simpul. Graf pada dasarnya menggambarkan keterhubungan antar simpul. Graf bisa digunakan untuk merepresentasikan peta, diagram, dan lain-lain.

B. Terminologi Graf

1. *Graf berbobot*
Graf yang sisinya memiliki harga.
2. *Graf berarah*
Graf yang beberapa atau semua sisinya memiliki arah. Sisi tersebut hanya bisa dilalui searah, dari simpul A ke simpul B, dan tidak bisa sebaliknya.
3. *Bersisian*
Sebuah sisi dikatakan bersisian dengan simpul jika simpul tersebut dihubungkan dengan sisi yang dimaksud.
4. *Bertetangga*
Sebuah simpul dikatakan bertetangga dengan simpul lain jika kedua simpul tersebut dihubungkan oleh satu sisi yang sama.
5. *Derajat*
Jumlah sisi yang bersisian dengan simpul.
6. *Lintasan*
Himpunan sisi $\{e_1, e_2, \dots, e_n\}$ di mana sisi e_i dan e_{i+1} bersisian dengan simpul yang sama untuk semua $1 \leq i \leq n - 1$.

7. Sirkuit

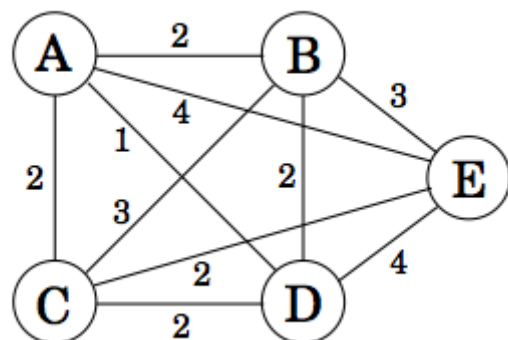
Lintasan yang simpul awal dan akhirnya sama.

C. Aplikasi Graf

Graf sering digunakan untuk menunjukkan keterhubungan antar benda. Contoh paling umum dari aplikasi graf adalah peta, di mana simpul menunjukkan kota dan sisi menunjukkan adanya jalan antar kota yang bersisian dengan sisi tersebut. Graf berbobot dapat digunakan untuk menggambarkan peta secara lebih detail. Harga pada sisi e menunjukkan panjang jalan antar simpul yang bersisian. Permasalahan ini menjadi terkenal melalui persoalan *The Travelling Salesman Problem* atau masalah pedagang keliling. Seorang pedagang ingin menjual barang dagangannya dengan cara mengunjungi setiap kota yang terdapat pada peta. Setiap kota memiliki jalan menuju kota lain dengan panjang jalan tertentu. Pedagang ini harus mengunjungi setiap kota dari kota tertentu dan kembali ke kota tersebut. Pedagang ini ingin melakukan hal tersebut sambil meminimalkan panjang jalan yang ditempuhnya.[2]

Dalam persoalan tersebut, setiap simpul menggambarkan kota dan sisi menyatakan bahwa terdapat jalan dari 2 kota yang dihubungkannya dengan panjang jalan sebesar harga sisi tersebut.

Graf berarah juga dapat menggambarkan peta jalan raya, di mana terdapat beberapa jalan yang satu arah. Sisi yang berarah menunjukkan jalan satu arah sementara sisi tak berarah menunjukkan jalan dua arah.



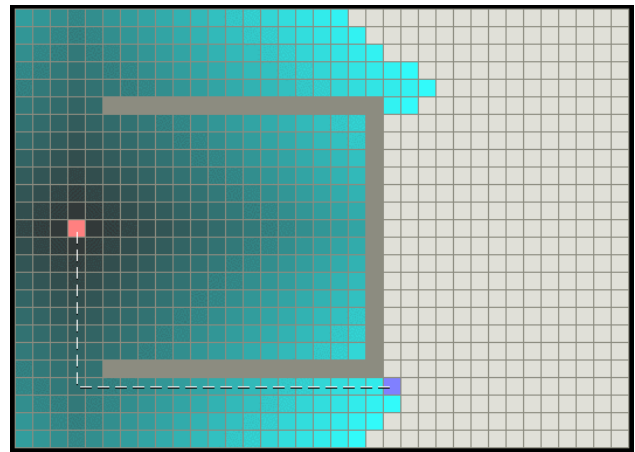
Gambar 2. Contoh soal *Travelling Salesman Problem*
Sumber: <https://www.quora.com/How-do-I-solve-the-traveling-salesman-problem-using-dynamic-programming>
7 Desember 2016

IV. ALGORITMA A*

Algoritma A* adalah salah satu algoritma pencarian jalan pada graf yang dikembangkan oleh Peter Hart, Nils Nilsson, dan Bertram Raphael dari Stanford Research Institute [3]. Algoritma A* adalah pengembangan dari algoritma *Dijkstra* dengan menggabungkan kelebihan dari algoritma *Greedy Best-First Search*.

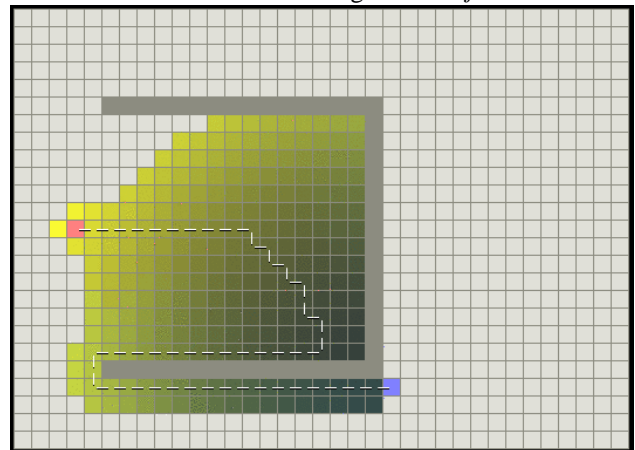
Algoritma *Dijkstra* bekerja dengan cara mengunjungi setiap simpul yang belum dikunjungi dan menentukan

harga setiap simpul, mulai dari simpul awal. Harga setiap simpul dihargai sebagai jarak terpendek dari simpul awal ke simpul tersebut. Kemudian, algoritma ini akan menelusuri simpul dengan harga terkecil dan mengulangi proses tersebut hingga akhirnya terbentuk sebuah pohon yang merepresentasikan jarak terpendek dari simpul awal ke semua simpul dalam peta.[4] Algoritma ini akan menghasilkan jalan terpendek. Namun, untuk peta yang besar, proses ini tidak efektif ketika kita hanya ingin mencari jalan menuju satu simpul tertentu, karena algoritma *Dijkstra* akan terus menelusuri setiap simpul hingga mencapai simpul yang dituju, walaupun simpul tersebut semakin menjauhi titik tujuan. Gambar 3a menunjukkan proses kerja algoritma *Dijkstra*.



Gambar 3a. Proses Algoritma *Dijkstra*

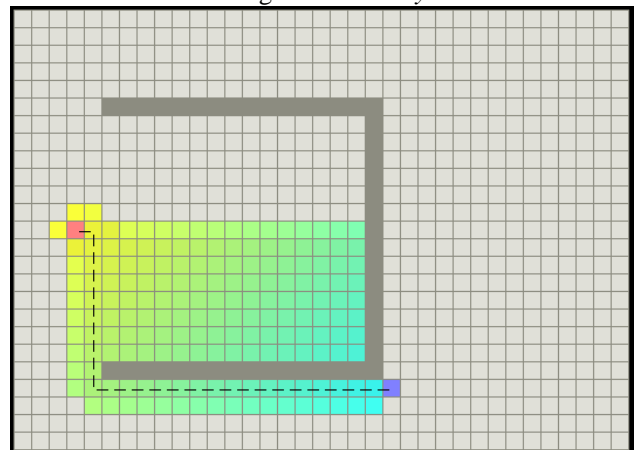
Algoritma *Greedy Best-First Search* bekerja dengan cara menjelajahi simpul yang semakin mendekati simpul tujuan dan menelusuri simpul tersebut.[4] Pada peta tanpa halangan, algoritma ini dapat bekerja dengan baik dan efektif karena jumlah simpul yang dijelajahi menjadi sangat sedikit. Namun, algoritma ini mengalami masalah jika terdapat penghalang antara simpul awal dan akhir. Algoritma ini akan terus mendekati simpul akhir meskipun terdapat penghalang. Akibatnya, algoritma ini akan menyediakan jalan yang memutar objek penghalang tersebut, bukan menyediakan jalan yang efektif. Gambar 3b menunjukkan proses kerja algoritma *Greedy Best-First Search*.



Gambar 3b. Proses Algoritma *Greedy Best-First Search*

Algoritma A^* menggabungkan kelebihan kedua algoritma tersebut. Algoritma A^* bekerja dengan cara berikut. Pertama, setiap simpul diberi nilai g yang merupakan “perkiraan” jarak menuju simpul akhir. Nilai ini akan membesar untuk simpul yang semakin jauh dari simpul awal.

Kedua, algoritma ini bekerja seperti algoritma *Dijkstra* dan menghargai setiap simpul dengan nilai f yang merupakan jarak sesungguhnya dari simpul awal. Namun, algoritma A^* akan menelusuri simpul yang memiliki harga $h = f + g$ yang terkecil, bukan hanya harga f seperti algoritma *Dijkstra*. [4] Jika nilai g ditetapkan dengan baik, maka algoritma A^* dapat mencari jalan terpendek dengan “mempertimbangkan” apakah simpul yang dipilih mendekati atau menjauhi simpul akhir, sehingga algoritma A^* tidak perlu menjelajahi simpul yang menjauhi simpul akhir. Gambar 3c menunjukkan proses kerja algoritma A^* .



Gambar 3c. Proses Algoritma A^*

Gambar 3. Kotak berwarna menunjukkan simpul yang diperiksa dan garis putus-putus menggambarkan jalan terpendek yang dihasilkan setiap algoritma

Sumber:

<http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html> 7 Desember 2016

V. REPRESENTASI PETA

Secara garis besar, terdapat 2 jenis representasi peta. Pertama adalah peta dunia yang dijelajahi. Peta ini merupakan dunia virtual yang dijelajahi pemain, baik

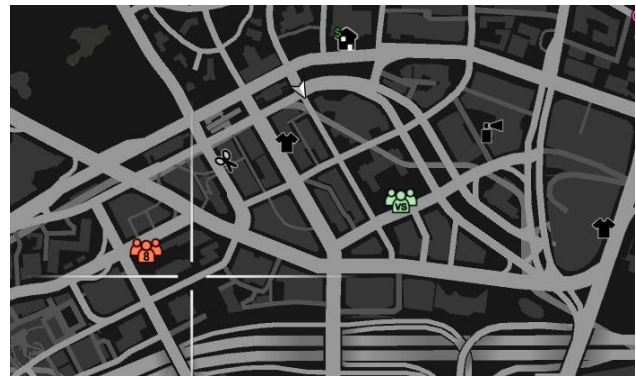
dalam 2D ataupun 3D. Kedua adalah peta navigasi. Peta jenis ini adalah penggambaran sederhana dari dunia virtual yang sesungguhnya. Peta jenis ini biasa digambarkan dalam 2D dan hanya menggambarkan jalan-jalan yang tersedia. Peta navigasi terdapat pada dunia sesungguhnya dalam bentuk GPS, Google Maps, dan aplikasi-aplikasi peta lainnya. Dalam makalah ini, kita akan membahas mengenai peta navigasi.

Peta navigasi dapat direpresentasikan dalam bentuk graf berbobot (lihat bab III GRAF). Graf jenis ini biasa disebut sebagai *waypoint graph*. Graf ini dibentuk dengan mengambil beberapa titik dalam peta sebagai simpul dan menghubungkannya. Biasanya, satu sisi menunjukkan bahwa terdapat jalan lurus bebas halangan dari satu simpul ke simpul satunya.[5] *Waypoint Graph* ini dapat dibentuk baik manual oleh pembuatnya maupun secara otomatis dengan algoritma. Kelebihan pembuatan graf secara manual adalah tidak akan terjadi kesalahan karena dibuat oleh pembuat permainan. Sementara kelebihan pembuatan graf secara otomatis adalah fleksibilitas graf jika suatu saat peta akan dirubah karena adanya *patch* atau tambahan dari pembuat permainan.

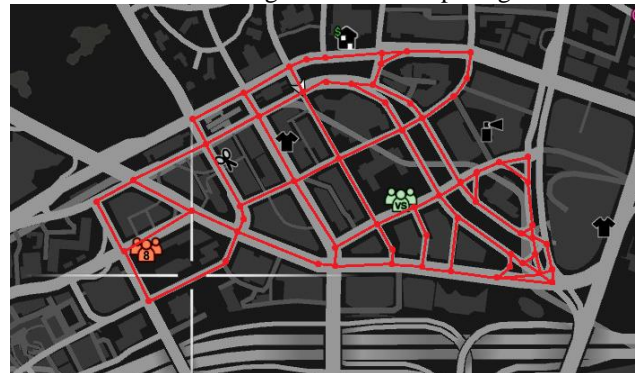
Waypoint Graph dalam peta navigasi akan menggambarkan secara garis besar jalan-jalan yang dapat dilalui. Dalam permainan *Grand Theft Auto V*, pemain dapat secara harafiah menjelajahi setiap sudut dunia. Namun, sistem navigasi hanya akan aktif jika pemain mengendarai kendaraan darat seperti mobil atau motor. Meskipun pemain sudah memasang titik tujuan, sistem navigasi tidak akan menunjukkan jalannya. Maka, dalam kasus ini, *waypoint graph* yang dibuat menggambarkan peta jalan raya yang dapat dilalui mobil. Pada gambar 4b, jalan raya yang dimaksud dilambangkan dengan warna abu-abu muda. Contoh *waypoint graph* yang dapat dibentuk digambarkan dalam gambar 4c.



Gambar 4a. Dunia virtual yang dijelajahi



Gambar 4b. Peta navigasi dari daerah pada gambar 4a



Gambar 4c. Contoh *waypoint graph* yang dapat dibentuk (warna merah)

Dalam permainan *Grand Theft Auto V*, terdapat beberapa jalan yang satu arah. Jalan satu arah ini dapat direpresentasikan dengan graf berarah (lihat bab III GRAF). Akibatnya, *waypoint graf* yang terbentuk memiliki beberapa sisi berarah.

VI. SISTEM NAVIGASI

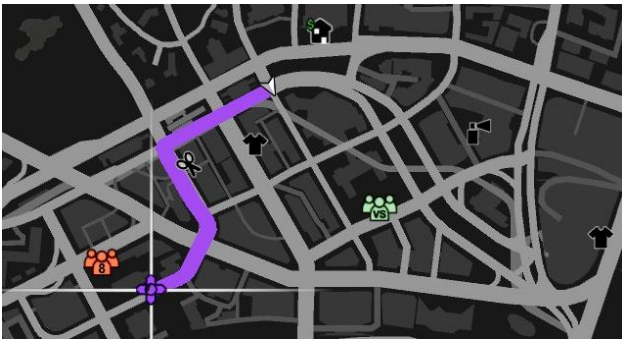
Bab ini akan mengambil contoh sistem navigasi yang terdapat dalam permainan *Grand Theft Auto*. Secara garis besar, ada 3 jenis titik tujuan dalam permainan ini, yaitu *objective point*, *quick gps*, dan sistem untuk memilih sendiri titik tujuan. *Objective point* merupakan titik tujuan yang otomatis muncul pada peta navigasi saat melakukan misi. Titik ini biasa berwarna kuning atau hijau. *Quick GPS* merupakan salah satu fitur dalam permainan ini. Pemain dapat memilih tempat-tempat tertentu yang disediakan seperti ATM, rumah, atau toko. Kemudian, permainan akan mencari lokasi tersebut yang paling dekat dengan pemain. Titik tujuan dari *Quick GPS* dan titik yang dipilih pemain berwarna ungu.

Sistem navigasi bekerja dengan cara sebagai berikut. Pertama, menentukan titik awal dan titik akhir. Titik awal merupakan posisi pemain. Titik akhir merupakan titik pada *waypoint graph* yang terdekat dari titik tujuan sesuai dengan yang sudah dijelaskan sebelumnya. Penentuan titik akhir ini dipilih agar dapat menangani kasus ketika pemain memilih titik yang tidak terdapat di jalan raya.

Setelah kedua titik tersebut ditetapkan, jalan tercepat ditetapkan dengan algoritma pencarian jalan. Salah satu

algoritma yang banyak digunakan adalah algoritma A* (lihat bab IV ALGORITMA A*). Algoritma tersebut akan menghitung jalan tercepat untuk sampai ke tempat tujuan pemain. Setelah dihitung, jalan tersebut akan ditampilkan pada peta kecil yang ada di ujung kiri bawah layar. Warna jalan mengikuti warna titik tujuan.

Ketika pemain bergerak, maka jalan yang ditampilkan juga harus diperbaiki karena posisi pemain berpindah. Jika pemain mengikuti jalan yang ditunjukkan, maka program dapat mempersingkat waktu dengan hanya menghilangkan jalan yang sudah ditempuh pemain. Tetapi jika pemain berjalan menuju tempat yang tidak ditunjukkan, maka program harus menghitung ulang jalan tercepat yang baru, karena bisa saja jalan yang lama tidak lagi menjadi jalan yang tercepat. Proses tersebut akan dilakukan terus hingga pemain membatalkan sistem navigasi, atau pemain sudah mencapai jarak tertentu dari tempat tujuan. Setelah itu, barulah sistem navigasi akan berhenti. Gambar 5 menunjukkan contoh jalan yang terbentuk saat sistem navigasi dijalankan.



Gambar 5. Contoh jalan yang terbentuk saat menjalankan sistem navigasi pada permainan *Grand Theft Auto V*.

VII. KESIMPULAN

Dalam pembuatan sistem navigasi, baik dalam permainan maupun GPS sesungguhnya, banyak menggunakan teori graf. Peta direpresentasikan sebagai graf dalam bentuk *waypoint graph*. Setelah menentukan titik awal dan titik tujuan, jalan terpendek dapat dicari dengan menggunakan algoritma, salah satunya adalah algoritma A*.

REFERENSI

- [1] "Definition of Open World, Free Roaming, and Sandbox". Dikutip 7 Desember 2016 dari <http://www.legenddiaries.com/special-features/difference-between-sanbox-and-open-world-games/>.
- [2] Hart, P. E., Nilsson, N. J., Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics SSC4*, 1968.
- [3] Patel, Amit, Introduction to A*. Dikutip 7 Desember 2016 dari <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>.
- [4] Geraerts, Roland, "Planning Short Paths with Clearance using Explicit Corridor". IEEE International Conference on Robotics and Automation, Mei 2010.

- [5] Jaklin, Norman, Geraerts, Roland, "Navigating through Virtual Worlds: From Single Characters to Large Crowds". Belanda: Utrecht University, 2015.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 7 Desember 2016

A handwritten signature in black ink, appearing to read 'Kevin Jonathan Koswara'.

Kevin Jonathan Koswara - 13515052