

Penentuan Rute Rotasi Tercepat Dalam Map Breeze Gim Valorant Menggunakan Algoritma Dijkstra

Farrel Farandieka Fibriyanto - 13520054¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13520054@std.stei.itb.ac.id

Abstract— Dalam permainan Valorant, waktu sangatlah penting. Apabila suatu pemain dapat menggunakan rute yang memakan waktu terkecil, pemain tersebut dapat memberikan kontribusi yang tinggi kepada tim. Rute dengan waktu terkecil tersebut dapat dicari menggunakan Algoritma Dijkstra di map gim yang sudah direpresentasikan dalam graf berbobot.

Keywords— Graf, Algoritma Dijkstra, Valorant, Breeze.

I. PENDAHULUAN

Valorant adalah permainan *First Person Shooter* yang dibuat oleh Riot Games untuk Microsoft Windows yang dirilis secara resmi pada 2 Juni 2020. Valorant mengambil inspirasi dari beberapa permainan *First Person Shooter* sebelumnya seperti seri Counter Strike.

Dalam permainan ini, akan terdapat 2 tim, tim *attacker* dan tim *defender*, dengan 5 pemain di tiap tim. Tugas utama tim *attacker* merupakan menanam “Spike” di salah satu situs peledakan. Tugas utama tim *defender* merupakan memastikan tim *attacker* tidak bisa melakukan tugas utama mereka.

Graf merupakan konsep matematika yang sering digunakan dalam ranah informatika untuk merepresentasikan objek-objek dalam suatu komputer. Suatu graf terdiri dari simpul-simpul (*vertices*) dan himpunan sisi (*edges*). Graf dapat dibagi dua berdasarkan bobotnya yaitu unweighted dan weighted graph. Dalam weighted graph, bisa diimplementasikan algoritma dijkstra untuk mencari jalur dengan bobot terkecil dari suatu simpul ke simpul tertentu.

Akan diperlukan suatu jalur dengan waktu terkecil untuk tim *defender* apabila tim *attacker* memutuskan untuk menyerang situs peledakan di sisi map yang lain, pemain *defender* perlu mencari rute terpendek untuk pindah dari tempat menjaga dia sebelumnya ke situs peledakan yang diserang tim *attacker*. Dalam makalah ini, akan dicari suatu jalur tercepat dari suatu lokasi menjaga tim *defender* ke situs peledakan yang lain di suatu map gim Valorant menggunakan algoritma Dijkstra.

II. LANDASAN TEORI

2.1 Graf

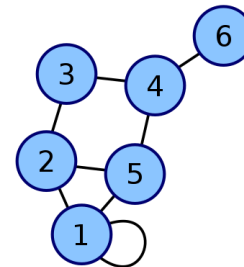
2.1.1 Definisi Graf

Graf merupakan tipe data abstrak yang digunakan untuk merepresentasikan hubungan kompleks non-linier antara objek.

Graf terdiri dari simpul (*vertices*) dan himpunan sisi antara dua simpul (*edges*). Secara matematis, graf dapat ditulis dalam notasi seperti berikut

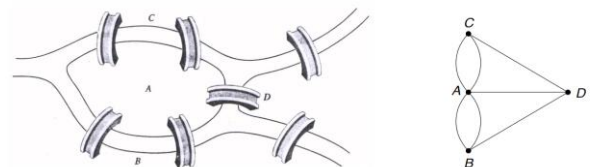
$$G = (V, E)$$

Dengan V merupakan himpunan tidak-kosong dari simpul (*vertices*) dan E merupakan himpunan sisi yang menghubungkan simpul (*edges*). Graf biasanya digambarkan berupa diagram, dengan lingkaran-lingkaran sebagai simpul dan garis di antara dua lingkaran sebagai himpunan sisi.



Gambar 2.1 contoh graf^[1]

Pada tahun 1736, Leonhard Euler membuat suatu makalah untuk menyelesaikan permasalahan Seven Bridge of Königsberg, makalah ini dianggap sebagai makalah pertama dalam penggunaan teori graf^[2].



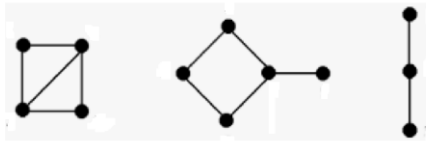
Gambar 2.2 permasalahan Seven Bridge of Königsberg dan graf yang setara dengan permasalahan tersebut^[3].

2.1.2 Jenis-Jenis Graf

Terdapat beberapa jenis graf yang perlu diketahui. Berdasarkan ada tidaknya sisi gelang atau sisi ganda, graf dapat dibagi menjadi dua, yaitu:

a. Graf Sederhana

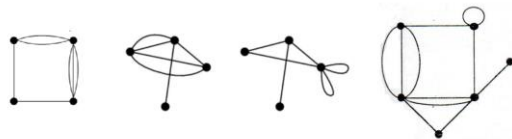
Tipe graf ini tidak mengandung gelang maupun sisi ganda sehingga disebut graf sederhana.



Gambar 2.3 contoh graf sederhana^[3]

b. Graf Tak-sederhana

Sisi gelang merupakan himpunan sisi yang menghubungkan suatu simpul dengan dirinya sendiri, sedangkan sisi ganda merupakan himpunan sisi yang menghubungkan dua simpul yang sama dengan salah satu himpunan sisi lainnya. Graf yang memiliki sisi gelang atau sisi ganda disebut graf tak-sederhana.

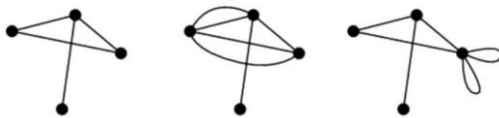


Gambar 2.4 contoh graf tak-sederhana^[3]

Berdasarkan orientasi arah pada sisi, graf dapat dibagi 2, yaitu:

a. Graf Tak-berarah

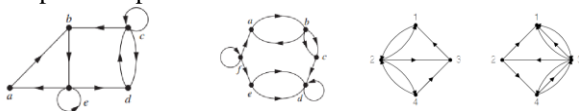
Graf yang tiap sisinya tidak mempunyai orientasi arah disebut graf tak-berarah



Gambar 2.5 contoh graf tak-berarah^[3]

b. Graf Berarah

Graf yang tiap sisinya mempunyai orientasi arah akan disebut graf berarah. Umumnya, orientasi arah pada sisi akan diilustrasikan dengan segitiga sehingga dalam sisi terdapat anak panah.



Gambar 2.6 contoh graf berarah^[3]

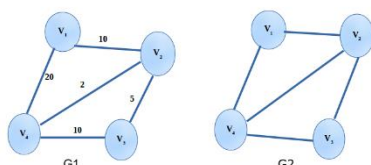
Berdasarkan bobot sisi graf, graf dapat dibagi 2, yaitu:

1. Graf Tak-berbobot

Graf yang tiap sisinya tidak diberi bobot biasa dipanggil graf tak-berbobot

2. Graf Berbobot

Graf yang tiap sisinya diberi bobot dipanggil graf berbobot



Gambar 2.7 contoh graf berbobot (G1) dan tak-berbobot (G2) ^[4]

2.1.3 Algoritma Dijkstra Dalam Graf

Algoritma Dijkstra merupakan salah satu algoritma yang dapat diterapkan dalam graf berbobot untuk menyelesaikan permasalahan rute terpendek. Algoritma ini diciptakan oleh Edsger W. Dijkstra pada tahun 1956, tetapi makalah tentang algoritma tersebut diterbitkan pada tahun 1959^[5]. Algoritma ini pada dasarnya akan melalui tahapan-tahapan seperti berikut^[6]:

1. Tetapkan simpul awal, yaitu simpul asal suatu rute, dan tetapkan jarak tentatif simpul tersebut dengan 0. Tetapkan jarak tentatif simpul lainnya dengan tak-hingga.
2. Tetapkan jarak suatu simpul ke simpul lainnya dengan memberikan bobot kepada tiap sisi.
3. Dari suatu simpul, yang dimulai dari simpul awal bila algoritma baru saja dimulai, pergi ke simpul lain yang dihubungi dengan suatu sisi. Hitung jarak tentatif simpul tersebut dengan menambahkan jarak tentatif simpul sebelumnya dengan bobot sisi, apabila jarak tentatif yang baru simpul tersebut lebih rendah daripada jarak tentatif simpul sekarang, maka tetapkan jarak tentatif tersebut sebagai jarak tentatif terbaru.
4. Lanjutkan tahapan ke-3 sampai semua sisi terlewati.

Setelah semua sisi terlewati, kita bisa mendapatkan rute terpendek dari simpul awal ke suatu simpul dengan melihat jarak tentatif simpul tersebut. Salah satu kelemahan algoritma ini merupakan kebutuhan algoritma untuk mengecek semua sisi dalam suatu graf sehingga mendapatkan rute terpendek untuk setiap simpul. Mungkin dalam beberapa aplikasi dibutuhkan rute terpendek untuk setiap simpul, tetapi pada umumnya kita hanya butuh rute terpendek dari suatu simpul ke suatu simpul, bukan ke semua simpul dalam graf.

2.2 Valorant

Valorant merupakan gim tembak-menembak daring yang diciptakan oleh Riot Games. Pada gim ini, pemain diberikan sejumlah uang untuk membeli senjata, dan menggunakan senjata tersebut untuk melaksanakan misinya. Misinya akan tergantung dari sisi mana pemain berada. Sisi *attacker* bertugas untuk meledakkan suatu situs dengan "Spike" yang diberikan di awal ronde, dan sisi *defender* bertugas untuk memastikan sisi *attacker* tidak bisa melaksanakan tugasnya. Ronde akan berakhir apabila semua pemain di sisi lawan sudah dieliminasi semua atau ketika "Spike" berhasil diredakan atau diledakkan. Tiap sisi akan memiliki 5 orang pemain sehingga permainan akan bertipe 5 lawan 5 dengan tim pertama yang mencapai 13 poin akan menang, kecuali dalam kondisi jus. Suatu tim yang terdiri dari 5 orang akan dipilih secara acak akan menjadi sisi *defender* atau *attacker* terlebih dahulu, ketika sudah melewati 12 ronde, tim akan ditukar sehingga apabila suatu tim menjadi sisi *defender* terlebih dahulu, setelah ronde 12 akan menjadi sisi *attacker*.

Agar semua situs peledakkan dilindungi, umumnya tim *defender* akan membagi tugas pemain mana yang akan melindungi suatu situs. Akan tetapi, belum tentu tim *attacker* akan menyerang semua situs, biasanya satu situs akan diserang oleh 5 orang pemain tim *attacker* sehingga pemain tim *defender* yang berusaha melindungi situs lainnya tidak bisa membantu anggota tim lainnya. Untuk mengantisipasinya, anggota tim

defender perlu pindah ke situs peledakan lainnya agar dapat membantu timnya. Dengan ini perlu dicari rute terpendek agar anggota tim *defender* bisa membantu anggota tim lainnya secepat mungkin.

Dalam gim ini, pemain dapat bergerak secara bebas secara planar menggunakan tombol w, a, s dan d di *keyboard*. Tergantung dengan rintangan dalam *map*, terdapat juga rintangan di mana pemain perlu melompat menggunakan *spacebar*, serta fitur-fitur yang dapat digunakan pemain menggunakan tombol f. Fitur-fitur dalam permainan akan tergantung dengan *map* yang sedang dimainkan. Beberapa contoh fiturnya merupakan “Door”, “Ropes”, dan “Teleporter”. Dalam *Valorant*, senjata yang sedang dipegang pemain berpengaruh dengan kecepatan berjalan maupun berlari pemain. Dengan senjata yang lebih berat, pemain akan berjalan dan berlari lebih pelan dibandingkan senjata yang lebih ringan. Apabila ingin berlari secepat mungkin, maka pemain perlu memegang pisau (*knives out*).

III. ANALISIS PERSOALAN

A. Penetapan Persoalan dan Metodologi Pengumpulan Data

Ditetapkan bahwa ada kebutuhan untuk menentukan rute terpendek dalam gim *Valorant*. Dalam makalah ini saya akan memilih *map Breeze* untuk dicari rute terpendeknya dari suatu situs peledakan ke situs peledakan lainnya. *Map* ini dipilih karena banyak pemain yang masih bingung rute mana yang tercepat.

Akan dibuat suatu graf sederhana tak-berarah berbobot dari *map Breeze* dari gim *Valorant*. Graf sederhana dipilih karena sisi dalam graf akan merepresentasikan jalan yang dapat diambil pemain, karena pemain tidak perlu berjalan-jalan dan kemudian balik ke tempat awal serta memiliki dua jalur untuk rute yang sama akan dipilih graf sederhana. Graf akan tak-berarah karena pemain tidak terbatas akan mobilitasnya sehingga apabila perlu melalui rute yang sama untuk arah sebaliknya diperbolehkan. Graf akan berbobot agar dapat ditentukan rute tercepatnya, dalam hal ini bobot sisi akan merepresentasikan waktu yang dibutuhkan tiap pemain untuk berlari dengan pisau (*knives out*) dari suatu simpul ke simpul lainnya.

Penentuan simpul dalam graf akan memperhitungkan beberapa faktor, seperti tempat mana yang biasanya didatangi pemain-pemain yang datang dari situs peledakan lainnya dan titik-titik mana saja yang biasa didatangi pemain-pemain karena lokasi strategisnya. Penentuan bobot dari suatu simpul akan dilaksanakan dengan menjalankan jam sukat dari suatu titik simpul di *map* dan berlari ke simpul lainnya, ketika sudah sampai di simpul tersebut mematikan jam sukat dan mencatat hasil waktu jam sukat.

B. Hasil Pengumpulan Data

Dari dalam gim, didapatkan susunan peta *map Breeze* sebagai berikut



Gambar 3.1 Susunan peta *map breeze* dari gim *Valorant*

Dengan templat gambar yang didapatkan penulis dari gim, akan dibuat graf sederhana menggunakan formula yang sudah diberikan di subbab A, dengan itu didapatkan hasil graf sebagai berikut



Gambar 3.2 Graf dari susunan peta *map Breeze*

Bila dilihat dengan saksama, terdapat satu sisi di graf yang memiliki arah meskipun penulis sudah menetapkan untuk membuat graf tak-berarah. Hal ini disebabkan dalam *map Breeze*, terdapat fitur “drop-down” atau “jatuh ke bawah” yang memungkinkan perjalanan satu arah. Sisi lain dalam graf akan tetap tak-berarah karena perjalanan dapat dilakukan ke dua arah.

Setelah mendapatkan graf, akan dicari bobot tiap sisi dengan metode yang sudah dijelaskan pada subbab A. Hasil pencarian bobot akan kemudian dituliskan di graf dan menghasilkan graf berikut



Gambar 3.3 Graf berbobot susunan peta *map* Breeze

Bobot beberapa sisi terlihat tidak sebanding dengan panjang sisinya dibandingkan dengan sisi lain. Hal ini disebabkan *map* ini tidak datar dan memiliki beberapa rintangan secara vertikal yang perlu di lewati pemain. Dalam *map* juga terdapat fitur “Ropes” atau semacam tali panjang untuk dipanjat pemain yang dapat mempercepat pemain mendapatkan elevasi dibandingkan dengan menggunakan “tangga” buatan. Di dalam gim, pemain juga bisa turun dengan metode melompat saja tanpa memakai “Ropes” yang sudah disebutkan tadi, metode ini juga lebih cepat untuk mengurangi elevasi. Apabila terdapat “Ropes”, akan dilakukan pengujian waktu dua arah untuk rute di sisi tersebut dan hasilnya akan dirata-ratakan agar dapat merepresentasikan kedua arah dengan baik. Di dalam *map* Breeze terdapat juga pintu besar yang terletak di “A Switch” dalam gambar 3.1. Diperlukan waktu untuk membuka pintu besar tersebut sehingga waktu yang dibutuhkan untuk melewati rute tersebut juga tidak sesuai dengan panjang sisi tersebut.

C. Penentuan Simpul Asal dan Simpul Tujuan

Sebelum diterapkan algoritma Dijkstra, akan ditetapkan simpul asal dan simpul tujuan agar kita tahu nilai jarak tentatif simpul mana yang merupakan hasil dari penerapan algoritma. Dalam hal ini, penulis akan menetapkan simpul di tengah situs sebagai simpul asal. Simpul tujuan akan berbeda-beda untuk tiap situs tergantung dengan titik temu pertama dengan tim lawan pada situs. Titik temu merupakan posisi dalam *map* yang pada umumnya memungkinkan pemain dari kedua tim untuk memiliki garis pandang lurus sehingga kedua pemain dapat saling menembak. Meskipun banyak tempat



Gambar 3.4 Graf berbobot susunan peta *map* Breeze untuk rotasi ke situs A (simpul merah sebagai simpul asal, simpul ungu sebagai simpul tujuan)

Apabila tim *attacker* memutuskan untuk menyerang situs peledakkan A, maka pemain di sisi *defender* yang berada di situs peledakkan B perlu berputar ke situs peledakkan A. Titik temu pertama pemain sisi *defender* yang berputar umumnya merupakan “Bridge”, “Metal Door”, “Double Door”, atau “A Main” sehingga akan ditetapkan simpul-simpul yang merepresentasikan titik tersebut sebagai simpul tujuan.



Gambar 3.5 Graf berbobot susunan peta *map* Breeze untuk rotasi ke situs B (simpul merah sebagai simpul asal, simpul ungu sebagai simpul tujuan)

Dan apabila tim *attacker* memutuskan untuk menyerang situs peledakkan B, pemain sisi *defender* yang berada di situs peledakkan A perlu berputar ke situs peledakkan tersebut. Umumnya pemain sisi *defender* akan bertemu pertama dengan pemain sisi *attacker* pada “Arches”, “Tunnel”, atau “B Main” sehingga akan ditetapkan simpul-simpul yang merepresentasikan titik-titik tersebut sebagai simpul tujuan.

D. Hasil Penerapan Algoritma Dijkstra

Setelah mendapatkan graf berbobot dan simpul awal dan tujuannya, kita bisa mengimplementasikan algoritma Dijkstra untuk menentukan rute terpendek. Akan terdapat perbedaan kecil untuk algoritma Dijkstra karena ada keberadaan satu sisi berarah, akan diterapkan algoritma Dijkstra yang menangani sisi berarah tersebut dan menerapkan algoritma normal untuk sisi yang lain.

Hasil penerapan variasi algoritma Dijkstra tersebut adalah sebagai berikut



Gambar 3.5 Graf berbobot susunan peta *map* Breeze hasil variasi algoritma Dijkstra untuk situs peledakkan A



Gambar 3.6 Graf berbobot susunan peta *map* Breeze hasil variasi algoritma Dijkstra untuk situs peledakkan B

Telah dihasilkan waktu terkecil yang mungkin untuk semua titik temu awal. Kemudian akan di-trace rute yang menghasilkan nilai tersebut.



Gambar 3.7 Graf berbobot susunan peta *map* Breeze hasil variasi algoritma Dijkstra untuk situs peledakkan A beserta rute yang diciptakan



Gambar 3.8 Graf berbobot susunan peta *map* Breeze hasil variasi algoritma Dijkstra untuk situs peledakkan B beserta rutenya.

E. Analisis Hasil Algoritma

Rute tercepat berhasil ditemukan untuk tiap situs peledakkan. Untuk situs peledakkan A, rute tercepat adalah melewati “Mid pillar” ke “Double Door” dengan waktu 14.5 detik. Untuk situs peledakkan B, rute tercepat adalah melewati “Mid pillar” ke “Tunnel” dengan waktu 16.5 detik. Untuk tiap situs, rute flank membutuhkan waktu terlama. Meskipun rute “Flank” untuk kedua situs mengambil waktu terlama, rute ini bisa membuat kaget sisi lawan dan memberikan kontribusi yang tinggi untuk tim. Semua gambar di atas dapat dilihat dengan resolusi tinggi dalam Arsip Pribadi Penulis^[7].

IV. KESIMPULAN

Mencari rute terpendek untuk rotasi pemain sisi *defender* dalam gim Valorant dapat dilakukan dengan merepresentasikan jaringan jalan yang bisa dilewati pemain dalam graf dan menggunakan algoritma Dijkstra untuk menentukan waktu

terkecil untuk mencapai titik temu lawan pertama. Dengan dilakukannya hal tersebut, pemain dapat menentukan rute tercepat yang bisa dilalui untuk memberikan kontribusi tertinggi pada tim. Akan tetapi, rute yang dibuat dalam makalah ini berasumsi bahwa semua kondisi sempurna sehingga rute terbaik pada ujungnya merupakan rute yang telah dipikir baik-baik oleh pemain tergantung dengan kondisi dalam permainan.

V. UCAPAN TERIMA KASIH

Puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmatnya dapat diselesaikan makalah ini dengan tepat waktu. Saya ingin mengucapkan terima kasih kepada dosen pengampu mata kuliah IF2120 Kelas 1 Bapak Rinaldi Munir atas ilmu yang diberikannya dalam kelas. Saya juga ingin mengucapkan terima kasih kepada keluarga dan teman-teman penulis yang telah mendukung penulis dalam menulis makalah ini.

REFERENCES

- [1] Situs Wikimedia, diakses 7 Desember 2021, <https://commons.wikimedia.org/wiki/File:6n-graph2.svg>
- [2] Biggs, N.; Lloyd, E.; Wilson, R. (1986), Graph Theory, 1736-1936, Oxford University Press
- [3] Munir, Rinaldi, diakses 11 Desember 2021, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/matdis21-22.htm>
- [4] Situs Codinggeek, diakses 11 Desember 2021, [4] <https://www.codinggeek.com/data-structure/graph-introductions-explanations-and-applications/>
- [5] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs" (PDF). *Numerische Mathematik*. 1: 269–271.
- [6] Gass, Saul; Fu, Michael (2013). Gass, Saul I; Fu, Michael C (eds.). "Dijkstra's Algorithm". *Encyclopedia of Operations Research and Management Science*
- [7] Arsip Pribadi Penulis, diakses 13 Oktober 2021, <https://imgur.com/a/53SafSt>

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.



Farrel Farandieka Fibriyanto
13520054

Bandung, 13 Desember 2021