

APLIKASI GRAF PADA *VIDEO GAME DOTA 2*

Iqbal Naufal / 13518074

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13518074@std.stei.itb.ac.id

Abstrak—Keberadaan video game telah ada sejak tahun 1996 dari sebuah game pada televisi yang bernama *Pong* ciptaan Ralph H. Baer. Seiring berkembangnya zaman, teknologi juga ikut berkembang. Dengan pesatnya perkembangan teknologi, perkembangan video game juga semakin cepat berkembang. Dimulai dari gambar yang hanya berbentuk 8 bit hingga sekarang yang bahkan bisa menyaingi dunia asli. Akhir-akhir ini video game tidak hanya digandrungi oleh usia remaja namun muda hingga tua. Video game juga sekarang tidak hanya dapat dimainkan sendiri namun dapat bermain secara asli dan tidak menggunakan *delay* dengan menggunakan jaringan internet. Video game juga telah diklasifikasikan sebagai cabang olahraga yang biasa disebut *E-Sport*. Salah satu *E-Sport* yang paling terkenal adalah sebuah game bernama *Dota 2*. Dengan menggunakan graf berbobot pemain dapat menentukan jalur-jalur yang paling efisien untuk menentukan tempat memperkuat diri dan menuju kemenangan.

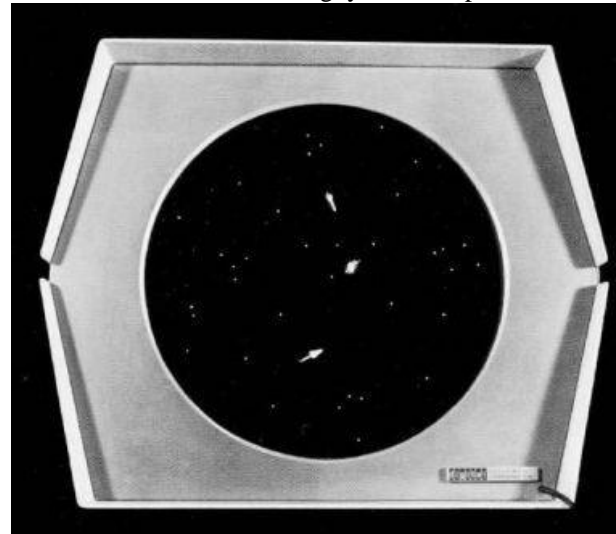
Kata Kunci—*Video game, E-Sport, Graf berbobot, Dota 2.*

I. PENDAHULUAN

Video Game merupakan sebuah istilah yang sudah tidak asing didengar sekarang ini. Permainan-permainan ini sangat digandrungi oleh berbagai macam kalangan dari bermacam-macam jenis usia. *Video Game* juga datang ke berbagai macam jenis alat-alat digital seperti gawai dan juga laptop. Pada saat ini *Video Game* dapat dimainkan kapan saja dan dimana saja. *Video Game* juga dimainkan dengan berbagai macam tujuan seperti sebagai hiburan dan bahkan hingga sebagai karir untuk menghasilkan uang. *Video Game* juga terdapat berbagai macam perlombaan atau yang biasa dikenal sebagai cabang olahraga *E-Sport*. *E-Sport* ini juga merupakan sebuah bentuk karir bagi sebagian orang yang telah memainkan permainannya sejak dulu maupun baru. Salah satu permainan dari *E-Sport* yang sangat terkenal adalah *Dota 2*. Dari permainan ini muncul berbagai macam atlet *E-Sport* yang kemudian menjadikan ajang lomba ini sebagai profesi karena hadiah dari lomba ini yang dapat dibidang sangat besar dan perlombaan yang tidak sedikit jumlahnya.

Video Game sendiri telah berada sejak tahun 1963 dimana saat itu terdapat permainan bernama *Spacewar!* oleh Steve Russel. *Spacewar!* telah dibuat untuk PDP-1 komputer mini oleh Digital Equipment dengan bantuan sebuah tim pengembang yang termasuk Martin Graetz, Pete Simson dan Dan Edwards. Mereka mengubah pola pikir masyarakat yang tadinya memandang komputer sebagai alat untuk bekerja saja menjadi sebuah alat serbaguna. Cara bermain *Spacewar!* Sangat simpel. Dua pemain mengontrol dua buah pesawat ruang angkasa yang kemudian saling menembaki satu sama lain dengan tujuan menghancurkan lawan. Dengan permainan yang

simpel ini, permainan ini menarik berbagai macam perhatian dari berbagai macam jenis khalayak umum. Namun terdapat beberapa kekurangan seperti misil yang ditembakkan akan selalu lurus akibat dari kurangnya mikroprosesor saat itu.



Gambar 1.1 Tampilan dari permainan *Spacewar!*

Sumber : kaskus.co.id

Lain halnya dengan penemuan konsol *Video Game* Pertama oleh Ralph H. Baer yang merupakan bapak konsol *Video Game* pertama di dunia. Ralph H. Baer lahir di Jerman, 8 Maret 1922. Ia tinggal di Amerika sejak usianya yang muda. Ralph merupakan seorang insinyur televisi yang saat itu masih sedikit jumlahnya. Selama pekerjaannya ia jalani, ia mengembangkan sebuah konsol permainan pada sebuah perusahaan bernama Sanders.

Konsol pertama ini kemudian dikembangkan yang kemudian menjadi prototip konsol game pertama yang dinamakan Brown Box yang telah dipatenkan pada tahun 1968. Kemudian paten ini dilisensi oleh Magnavox Odyssey yang kemudian menjadikannya menjadi konsol game pertama di dunia pada tahun 1972

Dengan besarnya kontribusi Ralph atas dunia teknologi, ia kemudian mendapatkan National Medal of Technology dari George Bush pada 13 Februari 2006 atas penemuannya yang mengawali dunia *Video Game*.



Gambar 1.2 Ralph H. Baer
Sumber : kaskus.co.id

Dalam perkembangan permainan, konsol di dunia juga semakin berkembang yang kemudian mulai merambah ke komputer dan menjadikan komputer sebagai salah satu konsol permainan yang paling umum digunakan di dunia saat ini. Salah satu permainan yang paling umum dimainkan adalah *Dota 2* yang menjadi salah satu cabang *E-Sport*. Permainan ini sangat menarik perhatian dari banyak orang karena jumlah hadiah dari perlombaan internasional yang resmi diadakan oleh *developer* dari permainan itu sendiri terbilang sangat banyak.

Dalam permainan *Dota 2* ini sendiri memiliki tujuan untuk menghancurkan *base* atau *ancient* lawan untuk memenangkan permainan ini. Permainan ini dimainkan oleh 10 orang yang terbagi menjadi 2 tim yang berisikan 5 orang. Selama permainan, pemain kemudian melakukan berbagai macam aksi untuk memperkuat karakternya yang kemudian akan digunakan untuk menahan serangan musuh dan menghancurkan *ancient* lawan. Aksi-aksi ini tidak terbatas dari mengumpulkan uang, membunuh musuh, ataupun melawan monster, namun juga sampai meng-*capture outpost* untuk mendapat poin untuk menaikkan level dari karakter hingga menaruh sebuah barang untuk melihat bagian dari peta permainan yang tidak terlihat oleh mata karakter untuk membantu teman ataupun diri sendiri melihat musuh yang datang ataupun perkembangan musuh.



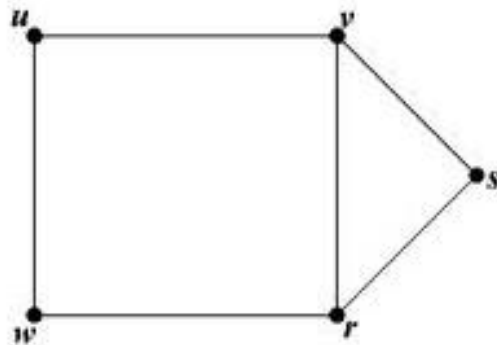
Gambar 1.3 Dota 2
Sumber : tekno.kompas.com

II. LANDASAN TEORI

2.1 Graf

Graf merupakan salah satu materi yang telah dipelajari pada mata kuliah Matematika Diskrit. Graf sendiri memiliki kegunaan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf dapat didefinisikan secara matematis sebagai pasangan dari himpunan simpul tidak kosong (V) dan himpunan sisi (boleh kosong) (E) dalam bentuk notasi Graf $G = (V, E)$. E merupakan himpunan sisi yang menghubungkan 2 buah simpul yang berada dalam himpunan simpul V .

Simpul-simpul biasanya disimbolkan dengan angka-angka (1, 2, 3, ...) atau huruf-huruf (a, b, c, ...), sedangkan sisi biasanya disimbolkan dengan huruf e (e_1, e_2, e_3, \dots) atau sebagai hubungan simpul didalam kurung seperti sisi yang menghubungkan simpul 1 dan simpul 2 dituliskan sebagai sisi (1,2).



Gambar 2.1 Contoh sebuah Graf
Sumber : dza42.blogspot.com

Pada contoh graf diatas merupakan himpunan $G = (V, E)$ dengan $V = \{u, v, s, r, w\}$ dan $E = \{(u, v), (v, s), (s, r), (v, r), (r, w), (u, w)\}$.

Graf dapat dikelompokkan dalam beberapa kelompok jenis graf dengan berbagai macam jenis sudut pandang dalam pengelompokkannya. Pengelompokkan jenis graf ini secara umum terdapat 2, yaitu berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, dan berdasarkan orientasi arah pada sisi.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:

1. Graf sederhana (*Simple graph*)

Graf sederhana merupakan graf yang dimana seluruh sisinya tidak memiliki sisi ganda (pada sepasang simpul hanya terdapat satu atau kurang sisi) dan tidak memiliki sisi gelang (simpul yang memiliki sisi dengan simpul itu sendiri). Pada graf sederhana tidak memiliki arah sehingga penulisan sisi (u, v) bernilai sama saja dengan sisi (v, u) .

2. Graf tak-sederhana (*Unsimple-graph*)

Graf tak-sederhana ini merupakan kebalikan atau lawan dari graf sederhana. Graf ini memiliki satu atau lebih sisi ganda atau sisi gelang ataupun keduanya didalam himpunan sisi dan simpulnya. Graf tak-sederhana terbagi lagi menjadi 2 jenis, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda merupakan graf yang memiliki setidaknya satu sisi gelang dan graf semu merupakan graf yang memiliki setidaknya satu sisi ganda. Jika graf memiliki kedua-duanya, graf tersebut masuk kedalam jenis graf semu.

Berdasarkan orientasi arah pada sisi, graf terbagi menjadi 2 jenis, yaitu :

1. Graf tak-berarah (*undirected graph*)

Graf tak-berarah merupakan graf yang dimana sisi dari suatu graf tersebut tidak memiliki arah sehingga penulisan sisi dari graf tersebut tidak berpengaruh urutannya. Sebagai contoh penulisan sisi (u, v) akan sama saja dengan (v, u) .

2. Graf berarah (*directed graph / digraph*)

Graf berarah merupakan sebuah graf yang dimana setiap sisi yang menghubungkan simpul-simpul tersebut memiliki arah. Sehingga urutan penulisan sisi akan berpengaruh dalam arti dari sisi tersebut. Sebagai contoh penulisan sisi (u, v) akan berbeda dengan sisi (v, u) .

Selain pengelompokkan jenis-jenis graf dari berbagai macam sudut pandang, terdapat juga beberapa elemen dari graf ini yang merupakan elemen-elemen penting, yaitu :

1. Ketetanggaan (*Adjacent*)

Ketetanggaan adalah sebuah elemen graf yang menyangkutkan dua buah simpul dimana dua buah simpul dapat dikatakan bertetangga apabila kedua simpul tersebut langsung terhubung oleh sebuah sisi.

2. Bersisian (*Incident*)

Bersisian adalah sebuah elemen graf yang berhubungan dengan sisi dan simpul. Sebuah sisi dapat dikatakan bersisian dengan sebuah simpul apabila sisi tersebut terhubung dengan simpul. Sebagai contoh apabila terdapat sisi (u, v) , maka sisi tersebut dapat dikatakan bersisian dengan simpul u atau simpul v .

3. Simpul Terpencil (*Isolated Vertex*)

Simpul Terpencil adalah sebuah elemen graf yang berhubungan dengan sebuah simpul. Sebuah simpul dapat dikatakan sebagai simpul terpencil apabila simpul tersebut tidak bersisian dengan sisi manapun atau dengan kata lain tidak berhubungan dengan simpul manapun.

4. Graf Kosong (*null graph* atau *empty graph*)

Graf kosong merupakan sebutan untuk graf dimana himpunan sisinya kosong atau dengan kata lain tidak terdapat satu pun sisi dalam graf tersebut. Dengan demikian, semua simpul dalam graf kosong dapat dikatakan sebagai simpul

terpencil.

5. Derajat (*Degree*)

Derajat merupakan sebuah elemen graf yang berhubungan dengan simpul dan sisi graf tersebut. Pada graf tak-berarah, derajat sebuah simpul dapat dikatakan sebagai jumlah sisi yang bersisian dengan simpul tersebut. Penulisan dari derajat sendiri adalah $d(V)$ dengan V adalah simpul. Sedangkan pada graf berarah derajat dibagi menjadi dua, yaitu derajat masuk suatu simpul dan derajat keluar suatu simpul. Derajat masuk adalah jumlah sisi yang bersisian dengan simpul tersebut dengan arah menuju simpul tersebut, sedangkan derajat keluar adalah jumlah sisi yang bersisian dengan simpul tersebut dengan arah keluar dari simpul tersebut. Derajat masuk suatu simpul dapat dituliskan sebagai $d_{in}(V)$ dan derajat keluar suatu simpul dapat dituliskan sebagai $d_{out}(V)$ dengan V adalah simpul tersebut.

6. Lintasan (*Path*)

Lintasan dengan panjang n dari sebuah simpul V_0 sampai dengan V_n adalah sebuah barisan simpul berselang sisi yang berbentuk seperti $V_0, e_1, V_1, e_2, V_2, \dots, V_{n-1}, e_n, V_n$ sedemikian sehingga $e_1 = (V_0, V_1), e_2 = (V_1, V_2), \dots, e_n = (V_{n-1}, V_n)$

7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Siklus atau sirkuit adalah sebuah lintasan yang berawal dan berakhir pada satu simpul yang sama. Jika sebuah lintasan berawal pada suatu simpul V_0 dan berakhir pada simpul V_0 maka lintasan tersebut merupakan sebuah siklus atau sirkuit

8. Terhubung (*Connected*)

Terhubung merupakan suatu sifat yang berhubungan dengan simpul dan graf itu sendiri. Sifat ini sangat mirip dengan ketetanggaan dengan perbedaan bahwa kedua simpul tidak harus terhubung dengan sebuah sisi saja. Jika terdapat suatu lintasan dengan awalan simpul pertama dan akhir simpul kedua, maka simpul pertama dan simpul kedua dapat dikatakan terhubung. Dan jika untuk seluruh kombinasi pasangan simpul yang terdapat dalam graf dapat dikatakan terhubung, maka graf tersebut dapat dikatakan sebagai graf terhubung. Jika graf merupakan graf berarah, graf dapat dikatakan graf terhubung dengan menghilangkan arahnya. Jenis terhubung ada dua jenis apabila graf merupakan graf berarah, terhubung kuat dan terhubung lemah. Apabila terdapat lintasan bolak-balik antara kedua simpul, maka dapat dikatakan bahwa kedua simpul terhubung kuat dan sebaliknya, jika hanya ada satu lintasan antara kedua simpul maka dapat dikatakan kedua simpul terhubung lemah.

9. Upagraf (*Subgraph*) dan komplemen upagraf

Misalkan $G = (V, E)$ adalah sebuah Graf. $G_1 = (V_1, E_1)$ merupakan upagraf dari G apabila V_1 merupakan himpunan bagian dari V dan E_1 merupakan himpunan bagian dari E .

Komplemen upagraf G_1 terhadap G adalah $G_2 = (V_2, E_2)$ dimana $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang bersisian dengan anggota-anggota himpunan E_2 .

10. Upagraf rentang (*Spanning subgraph*)

Upagraf rentang dari sebuah graf adalah upagraf dari graf tersebut yang mengandung semua simpul yang dimiliki graf tersebut. Penulisannya adalah upagraf $G_1 = (V, E_1)$ dari $G = (V, E)$.

11. *Cut-Set*

Cut-Set merupakan sebuah himpunan sisi dari sebuah graf terhubung dimana jika semua sisi dalam himpunan tersebut

dihapus dari graf terhubung tersebut maka graf menjadi graf tak-terhubung. Namun, sebuah himpunan sisi *Cut-Set* tidak dapat dikatakan sebagai *Cut-Set* apabila terdapat himpunan bagian dari himpunan sisi tersebut yang merupakan *Cut-Set*.

12. Graf berbobot (*Weighted graph*)

Graf berbobot adalah sebuah graf dimana setiap sisinya diberikan sebuah nilai yang menjadi harga atau bobot sisi tersebut yang akan berpengaruh dalam penggunaan graf tersebut.

2.2 Pohon

Pohon merupakan salah satu materi yang telah dipelajari dalam mata kuliah matematika diskrit setelah graf. Pohon sendiri merupakan materi percabangan dari graf sehingga semua pohon adalah graf, namun semua graf belum tentu pohon. Pohon didefinisikan sebagai graf tak-berarah yang merupakan graf terhubung, namun pada graf tersebut tidak boleh terdapat sirkuit atau siklus.

Agar dapat dikatakan sebagai pohon, sebuah graf tak-berarah sederhana $G = (V, E)$ dengan jumlah simpul n harus memenuhi beberapa sifat dibawah ini :

1. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
2. G terhubung dan memiliki $m = n - 1$ buah sisi.
3. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
4. G akan menjadi graf tak-terhubung apabila salah satu sisinya dihapus atau dengan kata lain semua sisi dapat dijadikan *Cut-Set*.

2.3 Pohon Merentang

Pohon merentang adalah sebuah upagraf merentang yang merupakan sebuah pohon dari sebuah graf terhubung yang setidaknya memiliki satu sirkuit. Pohon merentang didapatkan dari memutus sirkuit pada sebuah graf.

Pohon merentang dapat diaplikasikan sebagai metode untuk mencari sebuah bobot minimum dari sebuah graf berbobot G . Bobot minimum ini didapatkan dari bobot seluruh sisi pohon merentang yang didapatkan dari dua buah algoritma untuk mencari pohon merentang minimum, yaitu algoritma prim dan algoritma kruskal.

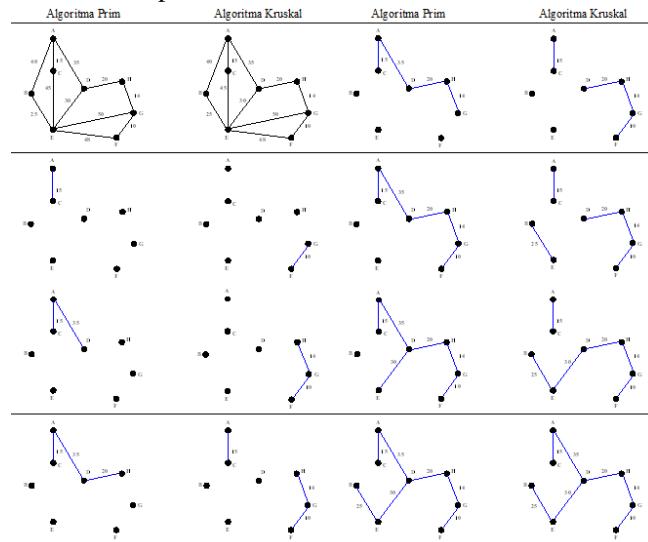
1. Algoritma Prim

Misalkan T adalah pohon merentang minimum pada suatu graf berbobot G . Algoritma prim adalah langkah-langkah untuk membuat pohon T tersebut agar menjadi pohon merentang minimum dari G . Langkah-langkahnya adalah sebagai berikut, pada setiap langkah diambil sisi graf G yang mempunyai bobot paling minimum yang bersisian dengan simpul yang sudah ada pada pohon T dan bila dimasukkan kedalam pohon T tidak akan membentuk sebuah sirkuit. Langkah ini kemudian diulangi terus sampai seluruh simpul graf G masuk kedalam pohon T .

2. Algoritma Kruskal

Algoritma Kruskal berfungsi sama seperti algoritma prim, yaitu untuk membuat sebuah pohon merentang minimum T dari sebuah graf berbobot G . Langkah-langkahnya adalah sebagai berikut. Pertama, seluruh sisi pada graf G diurut secara menaik berdasarkan bobot sisinya. Lalu sisi-sisi tersebut dimasukkan ke dalam pohon T sesuai dengan urutan pada urutan menaik yang

telah dibuat tersebut. Pemasukkan sisi ini dengan syarat bahwa jika sisi yang dimasukkan akan membentuk sirkuit maka sisi tersebut tidak valid dan tidak dimasukkan kedalam pohon T . Langkah ini kemudian diulang sampai seluruh simpul graf G masuk kedalam pohon T .



Gambar 2.2 Perbedaan Algoritma Prim dan Algoritma Kruskal

Sumber : dwipuspita53.wordpress.com

Dalam pembuatan pohon merentang minimum, hasil yang ditemukan tidak selalu unik. Sehingga bisa saja dengan menggunakan algoritma yang sama akan menghasilkan pohon merentang minimum yang berbeda. Ini bisa saja terjadi apabila terdapat sisi-sisi dengan bobot yang sama.

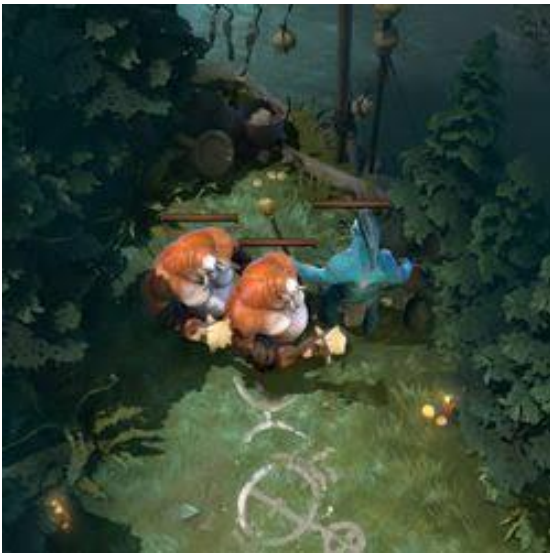
III. APLIKASI GRAF PADA VIDEO GAME DOTA 2

Penggunaan Graf pada *Dota 2* merupakan sebuah graf berbobot yang digunakan untuk menentukan jalur-jalur yang mungkin dilalui oleh karakter untuk melewati jalan yang terdapat pada peta permainan untuk menuju suatu tempat untuk menaikkan level dan mendapatkan uang atau akan disebut *camp* dari *camp* lain.

Simpul pada graf akan menggambarkan *camp-camp* yang berada di peta permainan. Pada graf kali ini akan diabaikan waktu untuk menghabiskan *camp* yang telah dilalui. Sisi pada graf akan menggambarkan jalur yang ditempuh oleh karakter untuk mencapai *camp* tertentu dari *camp* yang sedang ditempatinya, dimana bobot akan menggambarkan jarak dari jalur yang ditempuh tersebut.

Pewarnaan graf digunakan sebagai pembeda antara graf area *radiant* dan graf area *dire* yang merepresentasikan kedua tim dari permainan ini. Dibuat menjadi dua buah graf agar sesuai dengan permainan yang akan terjadi karena setiap pemain akan memperkuat karakternya pada areanya terlebih dahulu. Oleh karena itu, pemain harus bisa melewati semua simpul di graf area timnya terlebih dahulu.

Pada graf terdapat simpul yang berisi warna. Hal tersebut menggambarkan simpul awal atau posisi awal karakter untuk memulai perjalanannya.



Gambar 3.1 Contoh *Camp* pada *Dota 2*
 Sumber : dota2.gamepedia.com



Gambar 3.2 Peta permainan *Dota 2*
 Sumber : dota2.gamepedia.com

IV. PENCARIAN JALUR EFEKTIF DAN EFISIEN

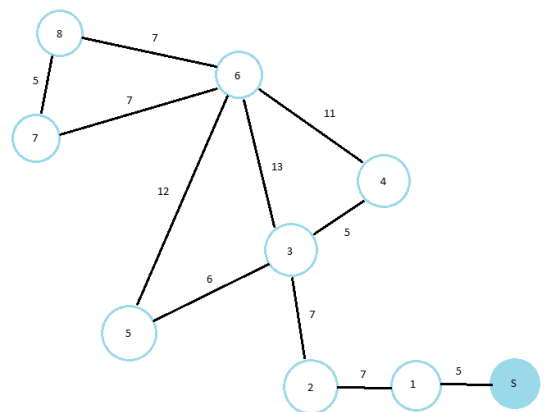
Untuk mencari sebuah jalur untuk memperkuat karakter yang efektif dan efisien dibutuhkan sebuah representasi graf dari peta tersebut. Selain itu juga dibutuhkan simpul awal yang akan menjadikan acuan dan membuat sebuah lintasan yang telah mengunjungi semua simpul pada graf atau *camp*. Selain itu juga akan dibutuhkan dua buah lintasan yang merepresentasikan pohon merentang minimum bagi kedua tim.

4.1 Peta Permainan dalam Graf

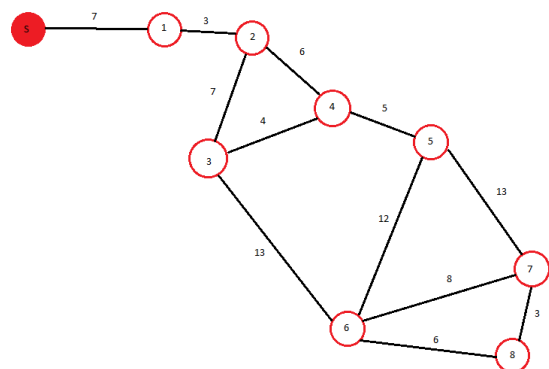


Gambar 4.1.1 Representasi Graf dari *Camp*
 Sumber : dota2.gamepedia.com

Pada gambar 4.1.1 menunjukkan sebuah peta permainan dari *game Dota 2* dimana untuk simpul biru menggambarkan area dari tim *radiant* dan simpul merah menggambarkan area dari tim *dire*. Simpul yang berisi menggambarkan simpul awal dari lintasan atau posisi awal karakter yang akan berjalan mengelilingi semua *camp* yang ada pada area timnya pada peta permainan. Representasi graf akan dibagi menjadi dua, yaitu representasi graf *radiant* dan graf *dire*.



Gambar 4.1.2 Representasi Graf berbobot *radiant*

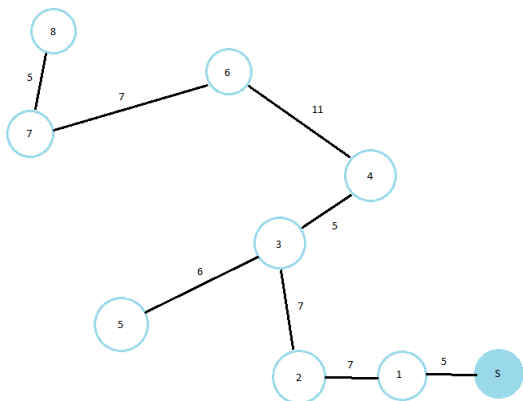


Gambar 4.1.3 Representasi Graf berbobot *dire*

Representasi graf berbobot diatas menggunakan karakter yang dapat terbang sehingga dapat melewati rintangan-rintangan yang menghalangi satu *camp* ke *camp* yang lain dan bobot merupakan jarak yang ditempuh karakter tersebut dengan cara terbang dan tidak terganggu oleh apapun seperti monster pada *camp* tersebut dan karakter musuh.

4.2 Menentukan Jalur Efektif dan Efisien tim *radiant*

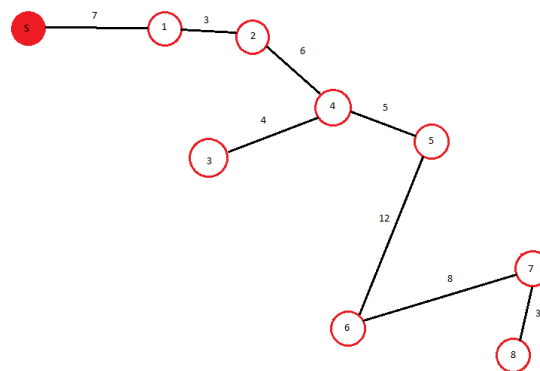
Dari gambar 4.1.2 telah terdapat graf berbobot yang terlihat sangat rumit. Cara mencari jalur yang efektif dan efisien ini dapat menggunakan algoritma Prim. Dengan menggunakan algoritma Prim, karakter pemain berangkat dari simpul S. Karakter kemudian mengambil jalur (S, 1) yang berbobot 5. Kemudian karakter mengambil jalur (1, 2) yang berbobot 7 sehingga total jarak menjadi 12. Kemudian karakter mengambil jalur (2, 3) yang berbobot 3 sehingga total jarak menjadi 15. Kemudian karakter mengambil jalur (3, 4) yang berbobot 5 sehingga total jarak menjadi 20. Kemudian karakter mengambil jalur (3, 5) yang berbobot 6 sehingga total jarak menjadi 26. Kemudian karakter mengambil jalur (4, 6) yang berbobot 11 sehingga total jarak menjadi 37. Kemudian karakter mengambil jalur (6, 7) yang berbobot 8 sehingga total jarak menjadi 45. Kemudian karakter mengambil jalur (7, 8) yang berbobot 3 sehingga total jarak menjadi 48. Dengan demikian, karakter telah mengunjungi seluruh *camp* dengan rute sebagai berikut bertotal jarak 48.



Gambar 4.2.1 Pohon merentang minimum tim *radiant*

4.3 Menentukan Jalur Efektif dan Efisien tim *dire*

Metode yang dilakukan sama seperti metode yang dilakukan pada perlakuan tim *radiant*. Menggunakan algoritma Prim juga dengan langkah-langkah sebagai berikut. Pertama karakter ambil sisi (S, 1) berbobot 7 sehingga total jarak menjadi 7. Kedua karakter ambil sisi (1, 2) berbobot 3 sehingga total jarak menjadi 10. Ketiga karakter ambil sisi (2, 4) berbobot 6 sehingga total jarak menjadi 16. Keempat karakter ambil sisi (4, 3) berbobot 4 sehingga total jarak menjadi 20. Kelima karakter ambil sisi (4, 5) berbobot 5 sehingga total jarak menjadi 25. Keenam karakter ambil sisi (5, 6) berbobot 12 sehingga total jarak menjadi 37. Ketujuh karakter ambil sisi (6, 7) berbobot 8 sehingga total jarak menjadi 45. Kedelapan karakter ambil sisi (7, 8) berbobot 3 sehingga total jarak menjadi 48. Hasil akhir dari rute perjalanan karakter dengan total jarak 48 adalah sebagai berikut.



Gambar 4.3.1 Pohon merentang minimum tim *dire*

V. KESIMPULAN

Pemain dapat mendapatkan *upper hand* dalam permainan saat pemain menggunakan jalur yang efektif dan efisien yang didapatkan dari pencarian rute menggunakan algoritma Prim ini. Dengan begitu, pemain akan membutuhkan waktu yang lebih sedikit dibandingkan dengan karakter tim lawan untuk menjadi lebih kuat dan dapat memenangkan permainan. Tentu saja ini bergantung pada cara bermain pemain. Jika pemain merupakan orang yang lebih berhati-hati dan mempersiapkan segala sesuatunya, jalur efektif dan efisien ini akan dipergunakan sehingga pemain dapat mempersiapkan terlebih dahulu apa saja yang akan memperkuat karakternya lebih cepat dibandingkan karakter lawan. Namun jika pemain lebih menyukai ikut dalam pertarungan maka akan merusak jalur akibat dari karakter pemain yang terus menerus berpindah tempat untuk membantu tim dan melawan musuh terlebih dahulu.

VI. UCAPAN TERIMA KASIH

Puji syukur penulis hadirkan kepada Allah SWT. Yang atas nikmat dan rahmat-Nya penulis dapat menulis dan membuat makalah ini. Penulis juga ingin mengucapkan rasa terima kasih sebesar-besarnya kepada ibu Dra. Harlili M.Sc yang atas kesabaran dan lelahnya lah saya bisa mendapatkan ilmu-ilmu mengenai Matematika Diskrit untuk menyelesaikan makalah ini dan masalah-masalah lain yang ada di kehidupan sehari-hari. Saya juga berterima kasih kepada kedua orang tua saya dan kakak saya yang karena doa-doa dan dukungannya lah saya terus bersemangat dalam menyelesaikan makalah ini.

REFERENCES

- [1] Munir, Rinaldi. Matematika Diskrit. 2015. [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf) (Diakses pada tanggal 5 Desember 2019 pukul 23.00)
- [2] Munir, Rinaldi. Matematik Diskrit. 2013. [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) (Diakses pada tanggal 5 Desember 2019 pukul 23.50)
- [3] Dwi, Susetyo. Video Game Pertama di Dunia. 2012. <https://techno.okezone.com/read/2012/02/13/326/574795/ini-dia-game-pertama-di-dunia> (Diakses pada tanggal 5 Desember 2019 pukul 21.00)
- [4] Jonathanna. Video Game dan Penemu Konsol Pertama di Dunia. 2013. <https://www.kaskus.co.id/thread/5145bce805346a9374000005/video-game-pertama-di-dunia-dan-penemu-konsol-game-pertama/> (Diakses pada tanggal 5 Desember 2019 pukul 19.00)

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, 5 Desember 2019



Iqbal Naufal
13518074