

Aplikasi Graf Berbobot pada Video Game FPS

Hilmi Naufal Yafie / 13517035
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
Email : 13517035@std.stei.itb.ac.id

Abstrak— Keberadaan *video game* pada saat ini sudah menjadi sebuah trending, baik dikalangan usia muda maupun tua. Pesatnya perkembangan teknologi beriringan dengan pesatnya perkembangan *video game*. *Video game* sendiri saat ini memiliki banyak genre guna menyesuaikan dengan keinginan para pemain, salah satu yang memiliki banyak peminatnya adalah *video game* bergenre FPS (*First Person Shooter*). Meskipun masih menuai kontroversi hingga saat ini, tidak bisa dipungkiri bahwa *video game* FPS memiliki peminat yang besar karena adrenalin yang dirasakan ketika bermain sulit didapatkan dari *video game* lainnya. Dalam *video game* FPS, kemampuan menembak saja tidaklah cukup, tapi strategi untuk menjelajahi area permainan juga merupakan hal yang penting. Dengan memanfaatkan graf berbobot, pengambilan jalur permainan yang efektif dan efisien dapat dilakukan sebagai salah satu kunci untuk mencapai kemenangan.

Kata Kunci — FPS, Graf Berbobot, *Video Game*.

I. PENDAHULUAN

Video game pada saat ini sudah menjadi suatu trending baru di masyarakat, terutama kalangan muda. Semakin pesatnya perkembangan teknologi digital saat ini membuat banyak perkembangan pada alat-alat digital, termasuk gawai atau sejenisnya yang juga bisa digunakan untuk bermain *video game*. Pada saat ini *video game*, yang dulunya hanya dimainkan oleh kalangan muda sebagai hiburan maupun pelepas penat, bisa menjadi salah satu pekerjaan yang dapat menghasilkan uang. Banyaknya perlombaan *video game* yang diadakan oleh perusahaan pengembang *video game*, sebagai salah satu bentuk promosi mereka, membuat para pemain *video game* saat ini bermain tidak hanya untuk sekadar hiburan lagi, tapi juga untuk menjuarai perlombaan yang diadakan tersebut. Bahkan, saat ini muncul istilah atlet eSport, yaitu atlet yang bermain dalam cabang *video game* tertentu untuk memperebutkan kejuaraan dengan nominal hadiah yang terbilang besar jumlahnya. Selain itu, *video game* juga bisa menjadi salah satu bahan yang sangat menarik bagi para *content creator* di internet.

Video game sendiri pertama kali diciptakan pada tahun 1972 oleh Ralph H. Baer. Ralph yang saat itu bekerja sebagai seorang insinyur kontraktor di Sander Associates mencoba untuk membuat sebuah *video game* yang dapat dimainkan dengan televisi, karena keberadaan komputer pada saat itu masih cukup mahal dan jarang. Proyek yang ia kerjakan sejak tahun 1966 itu akhirnya membuahkan hasil. Ralph merilis sebuah *video game* bernama Pong yang dapat dimainkan tanpa menggunakan komputer. *Video game* tersebut sempat menjadi legenda yang membuat anak-anak pada masa itu ramai mendatangi tempat yang menyewakan *video game* buatannya untuk bermain. Pada

tahun yang sama, ia juga berhasil membuat sebuah konsol *game* pertama yang diberi nama Magnavox Odyssey, yang mana konsol *game* serta *video game* ini nantinya akan menjadi cikal bakal perkembangan industri *video game* di dunia.



Gambar 1.1. Ralph H. Baer Bersama Penemuannya
Sumber: techno.okezone.com

Semenjak penemuan *video game* dan konsol *game* pertama, banyak perusahaan-perusahaan yang bergerak di bidang *video game* bermunculan, seperti Atari, Sega, Nintendo, dan masih banyak lagi. Namun, karena begitu pesatnya perkembangan *video game*, beberapa dari perusahaan yang bergerak di bidang *video game* tersebut pada akhirnya bangkrut atau diambil alih oleh perusahaan lain karena tidak bisa ikut bersaing dalam perkembangan *video game*.

Video game sendiri saat ini ada berbagai macam genre yang tersedia, sehingga para pemain dapat memilih untuk memainkan *video game* yang sesuai dengan genre yang disukainya. Salah satu dari genre *video game* yang digemari saat ini adalah *video game* bergenre *First Person Shooter*, atau biasa disingkat FPS.

Video game bergenre FPS merupakan *video game* dimana tampilan pada layarnya menggunakan perspektif orang pertama, sehingga pemain akan merasakan bagaimana rasanya jika ia yang menjadi pemeran utama dalam *video game* tersebut. Genre FPS ini sendiri biasanya digunakan untuk *video game* yang mengambil tema peperangan atau yang berhubungan dengan tembak-menembak, karena dengan menggunakan perspektif orang pertama, maka pemain dapat ikut merasakan bagaimana jika ia berada pada posisi dari karakter yang dimainkan. Dengan begitu, adrenalin pemain saat bermain akan lebih terpacu untuk tetap bertahan hidup dan memenangkan permainan disaat yang bersamaan. Hal inilah yang membuat *video game* bergenre FPS banyak diminati. *Video game* FPS pada dasarnya dapat terbagi menjadi dua mode utama. Mode pertama adalah *video game* FPS

yang mengutamakan permainan bersifat *multiplayer*, yaitu pemain dapat bermain bersama dengan orang lain, contohnya adalah game Counter Strike atau HALO dimana pemain dapat bermain baik dengan teman ataupun dengan orang asing yang tidak dikenal sebelumnya. Yang kedua mengutamakan *storyline*, yaitu jalan cerita dari *video game* tersebut, contohnya adalah Call of Duty atau Battlefield, dimana pemain akan diajak untuk bermain sekaligus menelusuri bagaimana cerita yang dibawakan oleh pengembang *video game* tersebut. Terdapat pula *video game* yang memberikan kedua hal tersebut kepada pemain, sehingga pemain tinggal memilih mana kah yang lebih mereka senang.

Dalam *video game* FPS yang menggunakan mode *storyline*, pemain tidak hanya disuruh untuk sekadar menembaki musuhnya agar dapat memenangkan permainan, namun pemain akan diminta menggunakan taktik yang bisa berbeda untuk setiap misi yang dijalani tergantung dari bagaimana kondisi dan perintah yang didapat pada misi tersebut. Pada mode *storyline* juga, permainan dapat di simpan secara manual oleh pemain atau secara otomatis ketika pemain berhasil sampai di *save point*, sehingga pemain dapat melanjutkan permainan di lain waktu, berbeda dengan mode *multiplayer* yang harus dimainkan sampai permainan benar-benar berakhir karena pemain secara langsung terhubung dengan pemain lain di dalam permainan.



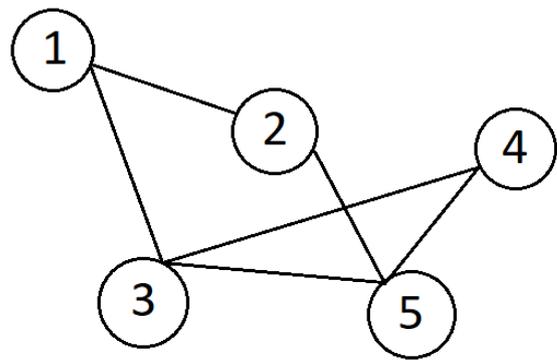
Gambar 1.2. Video Game Bergenre FPS
Sumber: *microsoft.com*

II. LANDASAN TEORI

2.1 Graf

Graf merupakan salah satu teori yang dapat dipelajari dari matematika diskrit, dimana graf sendiri secara matematis dapat didefinisikan sebagai berikut:

Suatu graf G yang merupakan pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang mana dalam hal ini, V merupakan himpunan yang tidak kosong dari simpul (*vertice* atau *node*) dan E merupakan himpunan sisi (*edge*), boleh kosong, yang menghubungkan dua buah simpul. Simpul pada graf biasanya dapat ditandai dengan simbol berupa huruf, seperti a, b, c, \dots atau dapat berupa bilangan seperti $1, 2, 3, \dots$ atau berupa gabungan keduanya. Sementara itu, sisi yang menghubungkan dua buah simpul, misal simpul a dan simpul b , dapat dinyatakan sebagai pasangan (a, b) atau dengan lambing e_i dengan i adalah sisi ke- i . Dengan contoh sebagai berikut:



Gambar 2.1.1. Contoh Sebuah Graf

pada contoh di atas, graf tersebut dapat didefinisikan dalam bentuk $G = (V, E)$ dengan $V = \{1, 2, 3, 4, 5\}$ dan $E = \{(1, 2), (1, 3), (2, 5), (3, 4), (3, 5), (4, 5)\}$.

Graf dapat dikelompokkan ke dalam beberapa jenis/kategori, bergantung pada sudut pandang dari pengelompokannya. Secara umum, graf dapat digolongkan menjadi dua kategori, yaitu graf sederhana (*simple graf*) dan graf tak sederhana (*unsimple graf*).

1. Graf Sederhana

Graf sederhana merupakan graf dimana sisi-sisi yang terdapat pada graf tersebut tidak memiliki sisi ganda (pada sepasang simpul terdapat lebih dari satu sisi) dan tidak memiliki sisi gelang (suatu simpul memiliki sisi dengan simpul itu sendiri). Selain itu, sisi pada graf sederhana merupakan pasangan tak terurut, sehingga penulisan sisi (u, v) , untuk sebuah simpul u dan simpul v , akan sama saja dengan (v, u) . Gambar 2.1 di atas merupakan contoh dari graf sederhana.

2. Graf Tak Sederhana

Graf tak sederhana merupakan kebalikan dari graf sederhana, yaitu suatu graf yang elemennya mengandung sisi ganda, sisi gelang, ataupun keduanya. Graf tak sederhana sendiri memiliki dua macam jenis, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda merupakan suatu graf yang mengandung setidaknya sebuah sisi ganda. Sementara, graf semu adalah sebuah graf yang mengandung setidaknya sebuah sisi gelang. Jika sebuah graf mengandung sisi ganda dan sisi gelang, maka graf tersebut termasuk ke dalam graf semu.

Jika melihat dari sudut pandang orientasi arahnya, maka graf dapat dibedakan menjadi graf berarah dan graf tak berarah.

1. Graf Berarah

Graf berarah merupakan suatu graf dimana sisi pada graf tersebut memiliki orientasi arah. Misal terdapat simpul u dan v serta sisi (u, v) , maka sisi tersebut dapat diartikan sisi dari simpul u menuju simpul v . Dengan begitu, posisi simpul dalam penulisan sisi pada graf berarah akan berpengaruh pada bentuk grafnya, dimana suatu sisi $(u, v) \neq (v, u)$.

2. Graf Tak Berarah

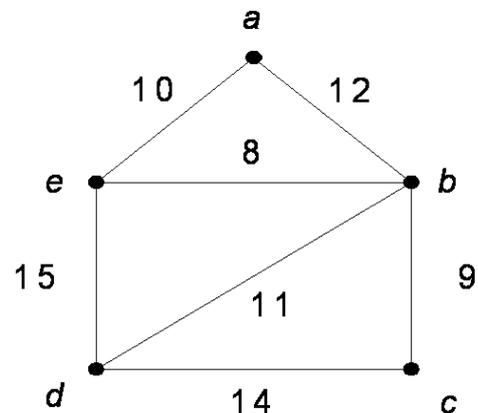
Graf tak berarah merupakan suatu graf yang tidak memiliki orientasi arah pada sisinya. Sehingga, bila terdapat suatu simpul u dan v pada graf tak berarah, maka penulisan sisi (u, v) dapat diartikan sebagai simpul u dengan simpul v dan $(u, v) = (v, u)$. Gambar 2.1 merupakan salah satu contoh graf tak berarah.

Selain sudut pandang orientasi yang membedakan jenis-jenis graf, terdapat pula beberapa istilah penting yang berkaitan dengan graf, yaitu:

- Bertetangga (*Adjacent*)**
Jika pada suatu graf tak berarah terdapat simpul u dan v dimana kedua simpul tersebut terhubung secara langsung oleh sebuah sisi (u, v) , maka dapat dikatakan bahwa simpul u dan simpul v bertetangga.
- Bersisian (*Incident*)**
Jika pada suatu graf terdapat sebuah sisi (u, v) , maka dapat dikatakan simpul u bersisian dengan simpul v dan sebaliknya.
- Simpul Terpencil (*Isolated Vertex*)**
Suatu simpul pada graf dikatakan simpul terpencil jika simpul tersebut tidak bersisian dengan satu pun dari simpul-simpul lain yang berada dalam graf yang sama.
- Graf Kosong (*Empty Graph*)**
Suatu graf dapat dikatakan sebagai graf kosong jika himpunan sisinya merupakan himpunan kosong, yaitu tidak ada sisi pada graf tersebut. Dapat pula dikatakan bila seluruh simpul yang ada pada graf kosong adalah simpul terpencil.
- Derajat (*Degree*)**
Pada graf tak berarah, derajat suatu simpul didefinisikan dengan banyaknya jumlah sisi yang bersisian dengan simpul tersebut. Derajat suatu simpul v dari graf tak berarah dinotasikan dengan $d(v)$. Sementara pada graf berarah, derajat suatu simpul dibagi menjadi dua, yaitu derajat simpul yang bersisian dengan sisi dengan arah menuju simpul dan arah dari simpul tersebut. Derajat suatu simpul v yang arahnya menuju v biasa dinotasikan $d_{in}(v)$, dan derajat yang arahnya dari simpul v biasa dinotasikan $d_{out}(v)$.
- Lintasan (*Path*)**
Suatu lintasan pada graf dengan panjang n dari sebuah simpul awal v_0 menuju simpul v_n merupakan barisan selang-seling simpul dan sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ dimana $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$.
- Siklus (*Cycle*) atau Sirkuit (*Circuit*)**
Jika lintasan pada suatu graf dengan simpul awal v_0 juga berakhir di v_0 , maka lintasan tersebut disebut sirkuit atau siklus.
- Terhubung (*Connected*)**
Jika pada suatu graf terdapat simpul u dan simpul v dimana lintasan dengan menggunakan simpul awal u dan simpul akhir v serta sebaliknya dapat terpenuhi, maka dapat dikatakan bahwa simpul u dan simpul v terhubung. Suatu graf disebut sebagai graf terhubung jika untuk setiap simpul u dan v yang ada pada graf, dapat terbentuk lintasan dari simpul u menuju simpul v . Bila tidak, maka graf tersebut disebut graf tak terhubung.
- Upagraf (*Subgraph*) dan Komplemen Upagraf**
Misalkan terdapat sebuah graf $G = (V, E)$ serta $G_1 = (V_1, E_1)$, graf G_1 dapat dikatakan sebagai upagraf dari graf G jika untuk setiap elemen $G_1, V_1 \subseteq V$ dan $E_1 \subseteq E$.

Sedangkan, komplemen dari upagraf adalah suatu graf $G_2 = (V_2, E_2)$ dimana $V_2 \subseteq V$ dan $E_2 \subseteq E$ dengan ketentuan $E_1 \cap E_2 = \{\}$ dan V_2 merupakan himpunan simpul yang bersisian dengan E_2 .

- Upagraf Rentang (*Spanning Subgraph*)**
Misalkan sembarang graf $G = (V, E)$ memiliki sebuah upagraf $G_1 = (V_1, E_1)$ dimana $V_1 = V$, maka upagraf G_1 dapat dikatakan pula sebagai upagraf rentang dari graf G .
- Cut-Set**
Jika suatu graf G adalah sebuah graf terhubung, dan terdapat E_1 , yang merupakan himpunan sisi, yang jika dihapus dari graf G akan menyebabkan graf G menjadi graf tak terhubung, maka E_1 dikatakan sebagai *cut-set*. Namun, bila terdapat himpunan E_2 dimana $E_2 \subset E_1$ dan jika E_2 dihapus dari graf G maka akan mengubah graf G menjadi graf tak terhubung, maka E_1 tidak bisa dikatakan sebagai *cut-set* dari graf G .
- Graf Berbobot (*Weighted Graph*)**
Suatu graf dikatakan sebagai graf berbobot jika setiap sisi dari graf tersebut memiliki nilai yang dapat berpengaruh ketika ingin menggunakan graf tersebut.



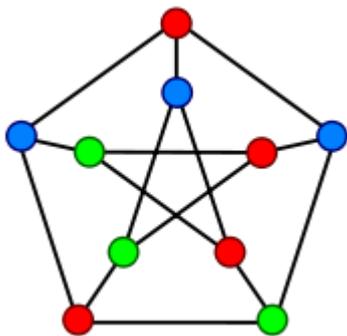
Gambar 2.1.2. Contoh Graf Berbobot
Sumber: sha-essa.blogspot.com

2.2 Pewarnaan Graf

Pewarnaan pada graf umumnya dilakukan untuk memetakan graf yang dimiliki agar terlihat lebih jelas. Ada tiga macam pewarnaan yang dapat dilakukan pada graf, yaitu pewarnaan simpul, pewarnaan sisi, dan pewarnaan wilayah (suatu bidang yang terbentuk oleh kumpulan sisi). Macam-macam pewarnaan dilakukan tergantung dengan bagaimana kebutuhan graf tersebut.

Pada pewarnaan simpul, pemberian warna bisa diartikan sebagai bentuk pembeda bagi suatu simpul dengan simpul lainnya dimana yang dibedakan bukanlah arti dari simpul tersebut, melainkan arti lain yang masih berhubungan dengan simpul tersebut. Sebagai contoh adalah pewarnaan peta, dimana pada pewarnaan peta bisa terdapat dua atau lebih wilayah dengan warna yang sama meskipun wilayah-wilayah tersebut saling berbeda satu sama lain. Pada pewarnaan peta, warna wilayah digunakan sebagai pembeda antar wilayah yang bersebelahan, sehingga mungkin saja terdapat wilayah yang

tidak bersebelahan dengan warna wilayah yang sama.



Gambar 2.2.1. Contoh Pewarnaan Simpul Graf
Sumber: dhukhasyamsy.blogspot.com

2.3 Pohon

Pohon merupakan salah satu teori yang dipelajari pada matematika diskrit, dimana pohon merupakan bentuk percabangan dari graf. Pohon didefinisikan sebagai suatu graf tak berarah dimana pada graf tersebut tidak ditemukan adanya siklus atau sirkuit. Dengan kata lain, setiap pohon dapat dikatakan sebagai sebuah graf, namun setiap graf belum tentu sebuah pohon.

Pohon memiliki sifat-sifat tertentu agar bisa dikatakan sebagai pohon. Misalkan $G = (V, E)$ merupakan suatu graf tak berarah sederhana dengan jumlah simpul n . Maka, sifat-sifat dari sebuah pohon G yang harus terpenuhi, yaitu:

- Setiap pasang simpul yang ada di dalam pohon G terhubung dengan lintasan tunggal.
- Pohon G merupakan graf terhubung dan memiliki m buah sisi, dengan $m = n - 1$.
- Pohon G tidak mengandung sirkuit.
- Setiap sisi dari pohon G dapat menjadi cut-set (jika sisi dihapus, maka graf akan terbagi menjadi dua buah komponen).

2.4 Pohon Merentang

Misalkan terdapat suatu graf $G = (V, E)$, dimana graf G adalah graf tak-berarah yang setidaknya memiliki sebuah sirkuit. Suatu pohon $T = (V_1, E_1)$ dapat dikatakan sebagai pohon merentang dari graf G jika pohon T merupakan upagraf dari graf G dan $V_1 = V$.

Salah satu dari penerapan pohon merentang adalah untuk mencari bobot minimum pada suatu graf berbobot G . Bobot pohon merentang suatu pohon T dari graf berbobot G didefinisikan sebagai jumlah seluruh bobot yang terdapat pada sisi pohon T . Pohon T dapat pula dikatakan sebagai sebuah pohon merentang minimum. Dalam mencari pohon merentang minimum, terdapat dua buah algoritma yang dapat dipakai, yaitu algoritma Prim dan algoritma Kruskal.

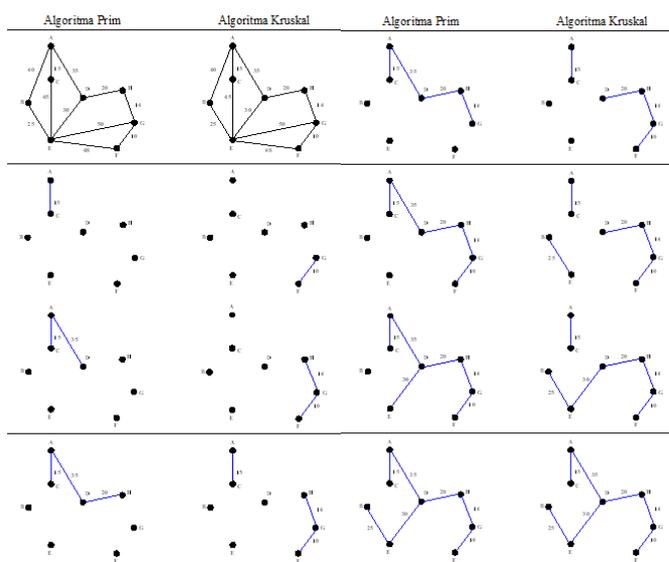
a. Algoritma Prim

Misalkan pohon T adalah pohon merentang minimum dari graf G , algoritma Prim digunakan untuk membentuk pohon T dengan langkah per langkah. Pada setiap langkah, dilakukan pengambilan dari sisi graf G

yang mempunyai bobot minimum dengan ketentuan sisi tersebut bersisian dengan simpul yang ada pada pohon T , dan bila sisi tersebut dimasukkan ke dalam pohon T tidak akan membentuk sirkuit. Proses ini diulangi hingga seluruh simpul pada graf G masuk ke dalam simpul pada pohon T .

b. Algoritma Kruskal

Algoritma Kruskal juga dapat digunakan untuk membangun sebuah pohon T , dimana T merupakan pohon merentang minimum dari graf berbobot G . Terdapat perbedaan pada cara algoritma Prim dan Kruskal. Pada algoritma Kruskal, seluruh sisi yang ada pada graf G diurut terlebih dahulu berdasarkan bobotnya. Lalu, sisi yang telah diurut dimasukkan ke dalam pohon T dengan ketentuan sisi tersebut tidak membentuk sirkuit di dalam pohon T . Langkah tersebut diulangi sampai seluruh simpul yang ada pada graf masuk ke dalam pohon T .



Gambar 2.4.1. Perbedaan Algoritma Prim dan Kruskal
Sumber: dwipuspita53.wordpress.com

Dalam pembuatan pohon merentang minimum, hasil yang didapat tidak selalu trivial, sehingga dimungkinkan untuk menemukan lebih dari satu pohon merentang minimum dengan nilai bobot yang sama.

III. APLIKASI GRAF PADA VIDEO GAME FPS

Graf pada dasarnya memiliki banyak aplikasi yang dapat diterapkan pada kehidupan sehari-hari, termasuk pada *video game* bergenre FPS. Pada *video game* FPS, graf diaplikasikan sebagai peta permainan dengan menggunakan graf berbobot dan pewarnaan graf. Namun terdapat sedikit perbedaan antara penggunaan graf pada mode *multiplayer* dan *storyline*.

3.1 Penerapan Graf pada Video Game FPS Mode Storyline

Pada *video game* FPS dengan mode *storyline*, graf berbobot digunakan pada peta permainan sebagai jalur yang harus dilalui oleh pemain untuk menyelesaikan suatu misi. Simpul pada graf

akan menggambarkan letak pada peta permainan dimana pemain tidak akan bertemu dengan musuh. Sebagian simpul juga dibuat sebagai *save point* yang secara otomatis akan menyimpan data permainan jika pemain telah mencapai posisi tersebut, sehingga ketika pemain kalah oleh musuh atau keluar dari permainan tanpa sempat menyimpan permainan, pemain tidak perlu mengulang permainan dari awal, namun bisa melanjutkan dari *save point* terakhir yang telah dicapai. Sisi pada graf digunakan sebagai penghubung antar simpul pada peta permainan, dimana bobot pada sisi graf menggambarkan musuh yang harus dihadapi untuk menuju ke tempat berikutnya.

Sementara, pewarnaan graf digunakan sebagai pembeda antara simpul-simpul yang memiliki *save point* dengan yang tidak, karena tidak semua simpul yang telah dilewati akan menyediakan *save point* agar permainan yang disajikan menjadi lebih menantang.

Untuk dapat menyelesaikan misi yang diberikan, pemain nantinya bisa diarahkan oleh NPC (*Non-Playable Character*) untuk melewati rintangan yang ada. Selain itu, NPC juga berperan dalam cerita-cerita pendek ketika permainan berlangsung untuk memperdalam cerita yang dimainkan. Pada misi dan kondisi tertentu, pemain juga dapat memilih sendiri jalur mana yang ingin dilewati dengan konsekuensi jika salah memilih, maka pemain akan menghadapi jalur dengan bobot yang besar, yang berarti pemain akan menghadapi musuh dengan jumlah yang lebih banyak.



Gambar 3.1.1. Player Mengikuti Petunjuk NPC untuk Menyelesaikan Misi
Sumber: thegurdian.com

3.2 Penerapan Graf pada Video Game FPS Mode Multiplayer

Pada *video game FPS* dengan mode *multiplayer*, graf berbobot juga digunakan dalam peta permainan sebagai penanda banyaknya musuh yang harus dilawan atau dilewati. Simpul pada graf menggambarkan area-area vital yang ada pada peta permainan, karena permainan hanya membutuhkan untuk mengalahkan lawan bermain, maka pemain tidak membutuhkan bantuan NPC dan dapat menjelajahi peta permainan sesuai keinginannya sendiri. Sisi pada graf merupakan penghubung antara simpul-simpul pada peta, dengan bobotnya merupakan musuh yang harus dihadapi untuk menuju daerah tersebut. Namun, karena musuh yang dilawan merupakan pemain lain yang juga dapat menggerakkan karakternya sesuai kemauan mereka, maka besaran bobot pada graf dapat berubah sewaktu-waktu, tergantung bagaimana kondisi permainan saat itu sedang berlangsung.

Sementara itu, pewarnaan graf pada *video game FPS* dengan

mode *multiplayer* tidak dibutuhkan karena tidak tersedianya fitur *save point* pada mode *multiplayer*, tidak seperti pada mode *storyline*.

Untuk dapat memenangkan permainan pada mode *multiplayer* akan sangat bergantung pada peraturan dari permainan tersebut. Ada peraturan yang mengharuskan untuk bertahan hingga akhir permainan, ada yang harus mendapatkan poin tertinggi dengan mengalahkan musuh sebanyak mungkin, ada yang saling merebut wilayah, dan masih banyak lagi. Namun pada intinya, pada permainan mode *multiplayer* pemain harus bisa bertahan selama mungkin dibanding para pemain lainnya untuk memperbesar kemungkinan memenangkan permainan.



Gambar 3.2.1. Salah Satu Video Game Mode Multiplayer
Sumber: halo.wikia.com

IV. MENENTUKAN JALUR EFEKTIF DAN EFISIEN

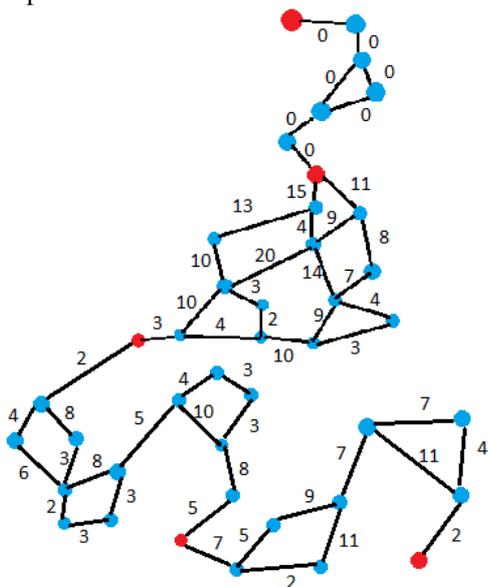
Untuk menentukan jalur paling efektif pada *video game FPS*, maka permainan haruslah memiliki simpul awal dan simpul akhir dalam permainannya. Dimana dalam mode *multiplayer*, tidak ada simpul akhir untuk mengakhiri permainan karena permainan pada mode *multiplayer* tidak memperdulikan dimana posisi terakhir pemain, namun bagaimana hasil akhir dari permainannya. Maka dari itu, hanya *video game FPS* dengan mode *storyline* yang dapat dicari jalur paling efektifnya.

4.1 Peta Permainan dalam Graf



Gambar 4.1.1. Peta Permainan Video Game FPS
Sumber: callofduty.wikia.com

Pada gambar 3.2.1. menunjukkan sebuah peta permainan dari salah satu *video game* FPS yang menggunakan mode *storyline*, dimana lingkaran kuning menandakan posisi awal pemain dan lingkaran merah menandakan posisi akhir permainan. Jika digambarkan dalam graf berbobot, maka peta permainan akan menjadi seperti berikut.



Gambar 4.1.2. Graf Berbobot dari Peta Permainan

Pada gambar 4.1.2., graf tersebut merupakan representasi dari peta permainan pada gambar 4.1.1. dimana simpul berwarna merah menandakan *save point* atau titik akhir dari permainan, sementara simpul berwarna biru menandakan percabangan jalan yang bisa ditempuh oleh pemain. Jika pemain kalah melawan musuh dan berada di simpul berwarna biru, maka pemain akan mengulang dari *save point* terdekat yang pernah dikunjungi sebelumnya. Sedangkan, bobot pada setiap sisi graf menandakan banyaknya musuh yang harus dihadapi jika ingin melewati sisi graf tersebut. Misi ini adalah salah satunya dimana pemain tidak mendapat bantuan dari NPC, sehingga keputusan ada di tangan pemain untuk memilih jalur mana yang paling baik.

4.2 Menentukan Jalur Efektif dan Efisien

Dari graf berbobot yang ditunjukkan pada gambar 4.1.2., jalur efektif dan efisien dapat ditentukan dengan mencari bobot terkecil yang dapat dilewati dari simpul awal hingga simpul tujuan. Dikarenakan adanya simpul yang berperan sebagai *save point* pada permainan, maka pencarian bobot minimum pada peta permainan dapat dilakukan dengan cara memecah pencarian dengan mencari bobot minimum yang dapat dicapai antar *save point* yang berdekatan.

a. Bobot Minimum antara *save point* pertama dan kedua
 Pada posisi diantara *save point* pertama dan kedua, bisa dilihat bahwa bobot graf yang ditunjukkan adalah nol, hal ini berarti tidak ada musuh yang berada di daerah itu, sehingga jalur manapun yang diambil oleh pemain untuk mencapai *save point* kedua tidak masalah. Pada bagian seperti inilah biasanya pengembang *video game* membangun jalan cerita permainan.

- b. Bobot Minimum antara *save point* kedua dan ketiga
 Dengan menerapkan algoritma Prim pada upagraf yang dibentuk antara *save point* kedua dan ketiga, dengan memulai dari simpul *save point* kedua, didapatkan bobot minimum yang diperlukan untuk menjelajahi area diantara *save point* kedua dan ketiga adalah 50. Terdapat dua jalur yang dapat dilewati oleh pemain sehingga jumlah musuh yang harus dihadapi menjadi minimum. Jalur pertama adalah melewati sisi, secara berurut, dengan bobot 15, 13, 10, 3, 2, 4, 3. Jalur kedua yang dapat dilewati oleh pemain adalah sisi, secara berurut, dengan bobot 11, 8, 7, 4, 3, 10, 4, 3. Meskipun kedua jalur tersebut memiliki jumlah bobot yang sama, perlu diperhatikan bahwa jalur pertama memiliki jumlah simpul yang lebih sedikit dibanding simpul kedua. Hal ini memberikan kelebihan dan kekurangan masing-masing pada kedua jalur. Pada jalur pertama, simpul yang harus diraih lebih sedikit dibanding jalur kedua, namun hal ini membuat jumlah musuh yang harus dihadapi setiap menuju simpul berikutnya menjadi lebih banyak. Jalur ini dapat menjadi pilihan bagi pemain yang ingin bermain cepat dan berlari menuju *save point* berikutnya tanpa melakukan banyak perlawanan terhadap musuh. Sementara itu pada jalur kedua, bobot yang harus dilalui untuk setiap simpul akan lebih sedikit dari jalur pertama, namun jumlah simpul yang harus ditempuh menjadi lebih banyak. Jalur ini dapat menjadi pilihan bagi pemain yang bermain secara hati-hati dan ingin menghadapi setiap lawan yang ditemui.
- c. Bobot Minimum antara *save point* ketiga dan keempat
 Dengan menggunakan algoritma Prim, bobot minimum yang diperlukan untuk menjelajahi area dari *save point* ketiga menuju *save point* keempat adalah 48. Jalur yang dapat dilewati sebenarnya hanya satu jalur utama. Namun, terdapat percabangan pada jalur tersebut, seperti pada sisi dengan bobot 2, 3, 3 dan sisi dengan bobot 8 atau sisi dengan bobot 4, 3, 3 dan sisi dengan bobot 10, dimana pada percabangan ini pemain disuruh untuk memilih jalur mana yang lebih cocok dengan cara bermain mereka. Sehingga, pemain dapat melewati jalur yang lebih cepat, yaitu sisi, secara berurut, dengan bobot 2, 4, 6, 8, 5, 10, 8, 5, atau jalur yang lebih aman, yaitu sisi, secara berurut, dengan bobot 2, 4, 6, 2, 3, 3, 5, 4, 3, 3, 8, 5.
- d. Bobot Minimum antara *save point* keempat dan terakhir
 Dengan menggunakan algoritma Prim, bobot minimum yang diperlukan untuk menyelesaikan permainan dari *save point* keempat adalah 40. Pada kasus ini, jalur yang dapat diambil merupakan sebuah jalur dengan sebuah percabangan di dalamnya dengan kasus yang sama seperti pada jalur *save point* ketiga menuju *save point* keempat. Jika pemain ingin melewati jalur dengan lebih cepat, maka pemain dapat melewati jalur dengan sisi, secara berurut, yang berbobot 7, 2, 11, 7, 11, 2. Namun, jika pemain ingin bermain lebih aman, maka dapat melalui jalur dengan bobot yang lebih tersebar, yaitu jalur dengan sisi, secara berurut, dengan bobot 7, 2, 11, 7, 7, 4, 2.
- Dengan menggabungkan kombinasi dari bobot minimum dari upagraf yang dibentuk dengan batas dua buah *save point* yang saling berdekatan, didapat bobot minimum yang

dibutuhkan untuk menyelesaikan permainan dari awal hingga akhir adalah 138. Dengan begitu, pemain dapat memenangkan permainan dengan menelusuri jalur yang efektif dan efisien, sehingga hanya akan menghadapi musuh dengan jumlah musuh paling sedikit 138.

V. KESIMPULAN

Pemain dapat memenangkan permainan *video game* FPS dengan mode *storyline* dengan melewati jalur yang efektif dan efisien yang didapat dari pencarian dengan menggunakan metode graf berbobot. Dengan begitu, pemain akan menghadapi musuh dengan jumlah minimum untuk memenangkan permainan. Jalur efektif dan efisien yang dapat dilalui juga tergantung dari bagaimana cara permainan yang digunakan oleh pemain. Jika pemain lebih cenderung untuk bermain cepat dan gesit, maka jalur yang dapat diambil adalah kombinasi jalur-jalur singkat dari masing-masing jalur yang tersedia, namun konsekuensinya adalah jumlah musuh yang dihadapi dalam satu waktu akan sangat banyak. Sementara, jika pemain lebih cenderung bermain secara perlahan dan berhati-hati, maka kombinasi jalur-jalur yang lebih panjang dari masing-masing jalur yang tersedia dapat menjadi pilihannya, karena musuh yang harus dihadapi akan lebih sedikit karena terbagi dengan banyaknya jalan yang dilalui, namun jarak yang ditempuh oleh pemain akan lebih jauh dan lama.

Untuk *video game* FPS dengan mode *multiplayer*, penggunaan graf hanya sebagai pemetaan posisi pemain lain. Pencarian jalur efektif dan efisien pada *video game* FPS dengan mode *multiplayer* tidak diperlukan karena untuk memenangkan permainan, pemain hanya harus mengikuti aturan yang diberikan dari *video game* tersebut.

VI. UCAPAN TERIMA KASIH

Puji dan syukur saya panjatkan kehadirat Allah SWT. karena berkat rahmat dan karunia-Nya, saya masih diberi kesempatan untuk membuat dan menyelesaikan makalah ini. Rasa terima kasih saya ucapkan kepada ibu Dra. Harlili M.Sc. beserta seluruh dosen mata kuliah IF2120 Matematika Diskrit yang telah memberikan ilmu pengetahuan dan bimbingan untuk membuat makalah ini. Saya juga berterima kasih kepada kedua orang tua saya yang selalu memberikan doa dan dukungan, beserta teman-teman yang memberikan bantuan kepada saya sehingga makalah ini dapat dibuat.

REFERENSI

- [1] Munir, Rinaldi. *Matematika Diskrit*. 2010. Informatika Bandung: Bandung, Edisi 3.
- [2] Hidayatullah, Madian. *Apa itu First Person Shooter?*. 2018. <https://esportsnesia.com/penting/apa-itu-fps/> (Diakses pada tanggal 7 Desember 2018 pukul 20.30)
- [3] Hedyanto, Abraham. *Sejarah Video Game: 10 Tahap Perkembangan dari Era Awal hingga Terkini*. 2018. <https://www.idntimes.com/tech/games/abraham-herdyanto/10-tahap-perkembangan-sejarah-video-game-dari-era-awal-hingga-sekarang-1> (Diakses pada tanggal 8 Desember 2018 pukul 8.00)
- [4] S., Simson P. *Kenalan Yuk Sama Penemu Konsol Game Pertama di Dunia*. 2017. <https://duniagames.co.id/news/1655-kenalan-yuk-sama-penemu-konsol-game-pertama-di-dunia> (Diakses pada tanggal 8 Desember 2018 pukul 15.25).

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, 10 Desember 2018



Hilmi Naufal Yafie
13517035