

# Aplikasi Graf dalam Penentuan Gerak NPC pada Permainan Counter-Strike

Arno Alexander - 13515141  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13515141@std.stei.itb.ac.id

**Abstrak** — Graf adalah sebuah struktur data yang mampu menggambarkan hubungan antara suatu objek dengan objek lainnya. Graf banyak diaplikasikan dalam berbagai hal di kehidupan sehari-hari. Contoh penerapannya dalam dunia permainan video adalah *navigation mesh* yang berfungsi untuk membantu *non-player character* (NPC) pada suatu permainan untuk bergerak dengan arah dan perilaku yang tepat. Dalam makalah ini, penulis akan mengulas *navigation mesh* pada salah satu permainan beraliran *first-person shooter* (FPS), yaitu Counter-Strike.

**Kata Kunci** — Counter-Strike, *first-person shooter*, graf, *navigation mesh*, *non-player character*.

## I. PENDAHULUAN

Perkembangan teknologi yang pesat memberikan pengaruh dalam banyak bidang, termasuk permainan. Sebelum komputer dan konsol permainan dikenal di masyarakat, permainan membutuhkan setidaknya komunikasi langsung antar pemain, interaksi dengan objek permainan yang nyata, atau aktivitas fisik. Saat ini, sudah ada jenis permainan yang tidak harus memenuhi salah satu dari ketiga hal tersebut, yaitu permainan video. Salah satunya objek fisik dalam permainan video adalah alat elektronik perantara yang digunakan untuk menerima masukan dari *controller* yang dikendalikan oleh pemain, mengolah masukan tersebut, dan menampilkan objek virtual dari permainan tersebut. Objek virtual tersebut adalah objek permainan sesungguhnya karena statusnya dipengaruhi oleh perilaku pemain.

Permainan video mulai terkenal sejak dikembangkannya permainan arkade. Kemudian, permainan video dapat diakses melalui konsol, komputer, hingga telepon seluler. Karena dapat diakses hampir di mana saja, konsumen permainan video semakin banyak dan heterogen sehingga aliran permainan video pun semakin beragam. Salah satu aliran yang terkenal adalah *first-person shooter* (FPS), yaitu permainan menembak dengan karakter yang dimainkan diposisikan sebagai orang pertama. Pemain seolah-olah diterjunkan ke dalam dunia permainan secara langsung karena sudut pandang tampilan permainan diposisikan sesuai dengan ruang pandang karakter.

Counter-Strike adalah contoh permainan video beraliran FPS yang dapat dimainkan baik dalam mode *single player*

maupun *multiplayer*. Terdapat empat seri utama, mulai dari Counter-Strike, Counter-Strike: Condition Zero, Counter-Strike: Source, hingga yang terakhir hingga sekarang adalah Counter-Strike: Global Offensive. Keempat seri tersebut dikembangkan oleh Valve Corporation dan beberapa pengembang lain. Untuk selanjutnya, penulis akan menggunakan nama Counter-Strike untuk merujuk pada seluruh seri utama permainan tersebut dengan penekanan pada Counter-Strike: Global Offensive.



**Gambar 1.** Counter-Strike: Global Offensive  
(Sumber: <http://store.steampowered.com/app/730/>  
diakses pada 3 Desember 2016)

## II. COUNTER-STRIKE

Sebelum membahas aplikasi graf pada Counter-Strike, penulis akan terlebih dahulu memaparkan deskripsi permainan tersebut pada bagian ini.

### A. Sejarah

Counter-Strike pertama kali muncul pada tahun 1999. Saat itu, Counter-Strike bukan sebuah permainan yang berdiri sendiri, namun merupakan modifikasi dari permainan FPS lain yang bernama Half-Life<sup>[4]</sup>. Modifikasi tersebut dikembangkan oleh Minh Le dan Jess Cliffe<sup>[5]</sup>. Modifikasi tersebut cukup digemari hingga akhirnya Valve Corporation memutuskan untuk merekrut mereka dan mengembangkan Counter-Strike sebagai permainan tersendiri. Dua versi pertama yang dirilis berturut-turut adalah Counter-Strike dan Counter-Strike: Condition Zero. Keduanya dikembangkan menggunakan *game engine* yang sama dengan Half-Life, yaitu GoldSrc. Counter-Strike pertama biasanya disebut Counter-Strike 1.6 untuk membedakannya dengan Counter-Strike secara umum.

Tahun 2004, Valve merilis *game engine* baru penerus GoldSrc, yaitu Source. Counter-Strike: Source adalah salah satu permainan yang dibuat menggunakan *game engine* tersebut dan menjadi yang pertama diterbitkan<sup>[6]</sup>. Versi Counter-Strike terbaru saat ini, Counter-Strike: Global Offensive yang diluncurkan pada 2012, juga dikembangkan di atas Source. Karena dibuat menggunakan *game engine* yang lebih modern, kedua seri terakhir ini memiliki kualitas yang lebih baik dalam hal grafis, suara, fisik, dan berbagai aspek lainnya dibandingkan seri pendahulunya.

## B. Gameplay

Inti dari Counter-Strike adalah pertarungan antar tim. Terdapat dua tim dalam permainan ini, yaitu *Counter-Terrorists* (CT) dan *Terrorists* (Teroris). Kedua tim tersebut memiliki misi yang berlawanan. Saat salah satu tim berhasil menjalankan misinya, tim yang bersangkutan memenangkan ronde tersebut dan tim lawan kalah. Kondisi menang dan kalah tidak hanya dapat dicapai melalui misi. Suatu tim juga bisa menang tanpa menjalankan misi dengan cara membunuh semua anggota tim lawan dengan syarat tim lawan juga belum menyelesaikan misinya. Untuk menentukan pemenang sesungguhnya, biasanya permainan dilakukan lebih dari satu ronde. Permainan baru akan selesai jika batasan jumlah ronde telah tercapai atau durasi bermain yang ditentukan habis. Tim yang menang di lebih banyak ronde akan dinyatakan sebagai juara.

Area virtual yang digunakan setiap tim untuk bertarung disebut *map* atau peta. Sebuah peta tersusun dari area-area yang memiliki nama tertentu. Pada awal ronde, setiap pemain akan dimunculkan pada suatu area di peta yang disebut *spawn zone*. Setiap tim memiliki *spawn zone* yang terpisah supaya tidak langsung terjadi kontak antar tim di awal ronde dan memberi keleluasaan sementara kepada setiap anggota tim untuk melakukan persiapan sebelum bertempur.

Setiap peta memiliki mode permainan yang unik. Mode permainan mempengaruhi misi untuk kedua tim. Counter-Strike memiliki banyak mode permainan, baik yang secara *default* ada dalam permainan maupun mode-mode hasil modifikasi para pemain. Terdapat dua mode yang selalu tersedia di dalam permainan sejak Counter-Strike 1.6 hingga Counter-Strike: Global Offensive, yaitu:

- **Bomb Defusal**

Pada peta yang memiliki mode ini, ada area khusus yang disebut *bombsite*. Teroris bertugas untuk menanam peledak C4 pada *bombsite*. Peledak tersebut memiliki hitungan waktu mundur. Jika teroris berhasil mengawalnya hingga hitungan waktu mundur habis, C4 akan meledak dan tim teroris dinyatakan memenangkan ronde.

Sebaliknya, tugas CT adalah mengawal *bombsite* agar teroris tidak menanam peledak hingga waktu ronde habis. Andai teroris berhasil menanam C4, CT harus berhasil menemukan dan menjinakkannya untuk menang.

Peta yang bermode *bomb defusal* memiliki awalan

‘de\_’ pada namanya, seperti *de\_dust*, *de\_dust2*, *de\_train*, dan lain-lain.



**Gambar 2.** Adegan teroris menanam bom<sup>[1]</sup>

- **Hostage Rescue**

Objek dari mode ini adalah salah satu jenis NPC, yaitu *hostage* (tawanan). Tawanan tidak boleh dilukai oleh tim manapun. Pemain yang melanggar peraturan tersebut akan mendapatkan hukuman berupa pengurangan uang atau bahkan dikeluarkan dari permainan. Uang yang dimaksud di sini adalah salah satu status pemain.

Posisi kedua tim pada mode ini dapat dikatakan berkebalikan dengan mode *bomb defusal* karena secara umum tugas teroris adalah mempertahankan dan tugas CT adalah menyerang. CT bertugas untuk menyelamatkan tawanan dari tempat semula ke area khusus di peta yang disebut *rescue zone* sebelum waktu ronde habis. Sebaliknya, tugas teroris adalah menjaga tawanan agar tidak diselamatkan oleh CT.

Nama peta yang bermode *hostage rescue* diawali dengan ‘cs\_’, misalnya *cs\_assault*, *cs\_italy*, *cs\_office*, dan lain-lain.



**Gambar 3.** Adegan CT menyelamatkan tawanan<sup>[1]</sup>

Status yang dimiliki pemain antara lain *health point* (HP), armor, uang, dan *inventory*. HP menyatakan jauhnya pemain dari kematian. Saat HP mencapai nol, pemain dinyatakan mati dan tidak dapat melanjutkan pertempuran hingga ronde selanjutnya dimulai. HP dapat berkurang akibat berbagai kejadian yang dialaminya, misalnya diserang pemain lain, jatuh dari ketinggian tertentu, atau terkena ledakan. Status armor menyatakan tingkat armor pemain. Ketika status armor tidak nol, pemain mendapatkan sedikit perlindungan dari serangan pemain

lain dan ledakan. Hasilnya, status armor akan berkurang dan HP pemain juga tetap berkurang meskipun lebih sedikit daripada saat tidak mengenakan armor. Uang digunakan dalam pembelian barang di *buy menu*, yaitu menu tempat pemain dapat membeli beberapa barang seperti senjata, armor, dan peralatan-peralatan lain. Barang yang dibeli akan masuk ke dalam *inventory*, yaitu daftar barang yang dimiliki pemain.

Pemain dapat membekali diri untuk pertarungan dengan mengakses *buy menu*. Menu tersebut hanya dapat diakses dari *buy zone*. Umumnya, *buy zone* terletak pada tempat yang sama dengan *spawn zone* agar pemain dapat segera membeli peralatannya sebelum bertempur. Setiap barang memiliki harga yang berbeda. Pemain hanya dapat membeli suatu barang jika uang yang dimilikinya cukup. Jika tidak, pembeliannya akan ditolak. Uang pemain akan bertambah setiap membunuh musuh, menjalankan misi, atau menyelesaikan ronde. *Buy mode* hanya dapat diakses selama durasi tertentu di awal setiap ronde. Saat durasi tersebut habis, seluruh pemain tidak dapat mengakses *buy mode* lagi hingga ronde selanjutnya dimulai.

### C. Antarmuka Pengguna (UI)

Kontrol pada permainan ini sama dengan sebagian besar permainan FPS. Pada *platform* PC, pengguna memerlukan *mouse* dan *keyboard* untuk mengontrol karakternya.

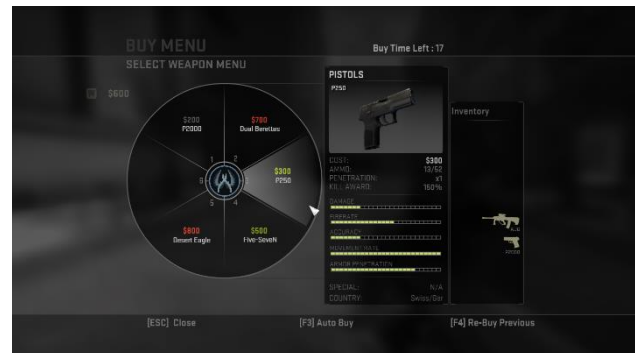


Gambar 4. Tampilan permainan [1]

Permainan memiliki tampilan seperti yang terlihat di gambar 4.

- Di bagian kanan bawah layar, terdapat informasi banyaknya sisa amunisi dari senjata yang sedang dibawa di tangan pemain. Tepat di atasnya, terdapat siluet dari seluruh senjata dan perlengkapan yang terdapat dalam *inventory*.
- Info mengenai HP dan status armor pemain terletak di bagian kiri bawah layar. Tampilan percakapan antar pemain dan pemberitahuan terletak tepat di atasnya.
- Pada bagian kiri atas layar, terdapat nama area yang sedang ditempati pemain, *minimap*, dan informasi uang. Bersama dengan nama area, *minimap* memberikan informasi mengenai lokasi pemain pada peta. Selain itu, *minimap* juga memberikan informasi tentang lokasi pemain lain, area khusus, dan objek-objek penting.

- Di tengah atas, ditampilkan sisa waktu ronde, skor setiap tim, dan avatar para pemain yang masih hidup dari tiap tim.
- Di kanan atas layar, terdapat informasi mengenai nama pemain yang baru saja terbunuh, senjata yang dipakai untuk membunuhnya, dan nama pembunuh.



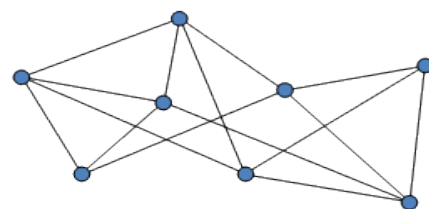
Gambar 5. Tampilan buy menu [1]

Saat pemain mengakses *buy menu*, akan terlihat jenis-jenis barang yang ditampilkan secara melingkar. Setelah memilih jenis barang yang akan dibeli, terlihat tampilan seperti pada gambar 5. Barang beserta harganya ditampilkan dalam lingkaran. Harga dapat ditampilkan dalam berbagai warna tergantung status pemain: abu-abu jika pemain sudah memiliki barang tersebut, merah jika uang pemain tidak cukup untuk membelinya, atau hijau jika uang pemain mencukupi. Saat *cursor* diarahkan ke atas suatu barang, akan ditampilkan statistik dari barang yang bersangkutan. Untuk membelinya, lakukan klik dan barang tersebut akan dimasukkan ke *inventory* yang ditampilkan di sebelah kanan *buy menu*.

## III. DASAR TEORI

### A. Graf

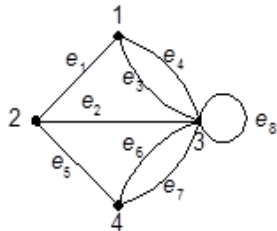
Graf adalah sebuah struktur diskrit yang terdiri dari simpul (*vertex*) dan sisi (*edge*). Dengan demikian, sebuah graf  $G$  dapat didefinisikan sebagai  $G=(V,E)$  dengan  $V$  adalah himpunan tak kosong dari simpul dan  $E$  adalah himpunan dari sisi<sup>[3]</sup>. Sisi dapat dituliskan sebagai  $(v_1, v_2)$  dengan  $v_1$  dan  $v_2$  masing-masing adalah dua buah simpul yang dihubungkan oleh sisi yang bersangkutan. Kegunaan graf adalah menggambarkan hubungan antar objek diskrit<sup>[2]</sup>. Simpul menyatakan objek, sedangkan sisi menyatakan hubungan antara objek yang dihubungkannya.



Gambar 6. Contoh graf dengan 8 simpul dan 15 sisi (Sumber: <http://www.uh.edu/engines/epi2467.htm> diakses pada 6 Desember 2016)

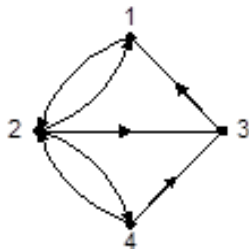


Dua buah simpul dapat dihubungkan oleh lebih dari satu buah sisi. Sisi itu disebut sisi-ganda. Contohnya adalah sisi  $e_6$  dan  $e_7$  pada gambar 7. Sebuah sisi dapat berujung pada simpul yang sama. Sisi itu disebut gelang. Sisi  $e_8$  pada gambar 7 adalah contoh gelang. Jika dalam suatu graf tidak terdapat sisi-ganda maupun gelang, graf tersebut disebut graf sederhana. Sebaliknya, graf tersebut adalah graf tak-sederhana. Graf tak-sederhana dapat dibagi menjadi dua jenis, yaitu graf ganda (memiliki sisi-ganda) dan graf semu (memiliki gelang).



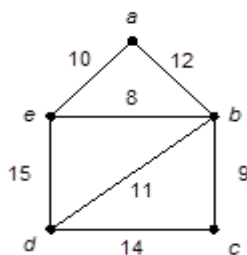
Gambar 7. Graf tak sederhana [2]

Tidak semua hubungan antar objek diskrit bersifat dua arah. Hubungan mungkin juga bersifat satu arah. Untuk merepresentasikannya, dibutuhkan sisi yang memiliki arah. Graf yang memiliki sisi berarah disebut graf berarah. Sisi pada graf berarah dapat disebut busur (*arc*). Sebaliknya, jika sisi graf tidak memiliki arah, graf tersebut disebut graf tak-berarah.



Gambar 8. Graf berarah [2]

Selain arah, suatu nilai juga dapat dipasangkan pada sisi graf untuk menyatakan harga dari sisi tersebut. Graf yang demikian disebut graf berbobot (*weighted graph*).



Gambar 9. Graf berbobot [2]

Terdapat beberapa istilah yang sering digunakan dalam teori graf, yaitu [2][3]:

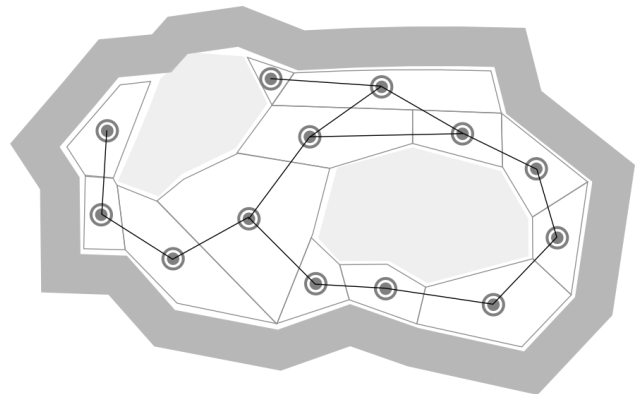
- Jika dua buah simpul pada graf tak-berarah terhubung secara langsung oleh sebuah sisi, kedua simpul

tersebut bertetangga (*adjacent*).

- Sisi  $e = (v_1, v_2)$  bersisian (*incident*) dengan simpul  $v_1$  dan  $v_2$ .
- Setiap simpul pada graf tak-berarah memiliki derajat, yaitu banyaknya sisi yang bersisian dengan simpul tersebut. Pada graf berarah, terdapat dua jenis derajat untuk sebuah simpul, yaitu derajat-masuk yang menyatakan banyaknya busur yang berakhir di simpul tersebut dan derajat-keluar yang menyatakan banyaknya busur yang berawal dari simpul tersebut.
- Misalkan terdapat sebuah graf  $G=(V,E)$ ,  $\{v_1, v_2, \dots, v_n\} \subseteq V$ ,  $\{e_1, e_2, \dots, e_{n-1}\} \subseteq E$ , dan  $e_1 = (v_0, v_1)$ ,  $e_2 = (v_1, v_2)$ ,  $\dots$ ,  $e_n = (v_{n-1}, v_n)$ . Barisan simpul dan sisi yang berurutan dan saling terhubung  $v_0, e_1, v_1, e_2, \dots, e_n, v_n$  disebut lintasan (*path*) dengan  $v_0$  sebagai simpul awal dan  $v_n$  sebagai simpul tujuan.
- Lintasan yang memiliki simpul awal dan simpul akhir yang sama disebut sirkuit (*circuit*).
- Jika dalam sebuah graf terdapat lintasan antara dua simpul, kedua simpul tersebut dikatakan terhubung.
- Graf  $G_s = (V_s, E_s)$  disebut upagraf (*subgraph*) dari  $G = (V, E)$  jika  $V_s \subseteq V$  dan  $E_s \subseteq E$ . Jika  $V_s = V$ ,  $G_s$  dapat juga disebut upagraf merentang (*spanning subgraph*) dari  $G$ .

### B. Navigation Mesh

*Navigation mesh* adalah sebuah struktur data yang terdiri dari poligon konveks yang saling bersinggungan dan terletak dalam sebuah peta. Gabungan seluruh poligon tersebut menggambarkan area yang dapat dilalui oleh NPC [8][9].



Gambar 10. Contoh navigation mesh dan representasinya dalam graf

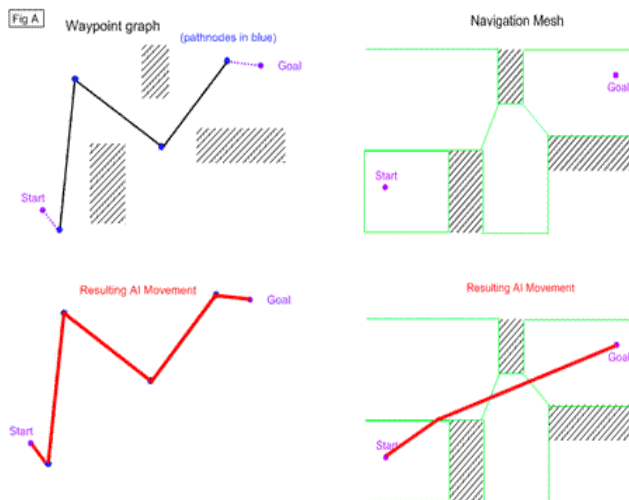
(Sumber: <http://jceipek.com/Olin-Coding-Tutorials/pathing.html> diakses pada 6 Desember 2016)

Dalam representasi graf, poligon-poligon penyusun *navigation mesh* dapat dimisalkan sebagai simpul. Sisi graf menyatakan hubungan antar poligon yang bersinggungan.

Fungsi *navigation mesh* adalah membantu NPC menentukan arah perjalanannya di peta. Saat suatu NPC menerima sinyal untuk bergerak dari suatu tempat ke tempat lainnya, akan dijalankan proses pencarian jalan terpendek dari area poligon tempat NPC semula ke area

poligon tempat titik tujuan berada. Proses tersebut disebut *pathfinding*. Ada berbagai algoritma *pathfinding* yang dapat dipakai, misalnya algoritma Dijkstra atau algoritma A\* (*A-star*)<sup>[10]</sup>.

*Navigation mesh* memiliki keunggulan dibandingkan metode lain yang memiliki fungsi yang sama, misalnya *waypoint graph*, yaitu representasi area yang dapat dilalui pada peta dengan menggunakan sejumlah titik yang saling berhubungan dengan titik-titik lain di dekatnya. Setiap titik mewakili area kecil di sekitarnya. Perpindahan NPC harus dilakukan dengan gerakan lurus dari suatu titik ke titik tetangganya, kemudian berbelok secara tiba-tiba, bergerak lurus lagi ke arah titik selanjutnya, dan seterusnya hingga mencapai titik tujuan. Hal tersebut dianggap kurang realistis dan tidak menyerupai gerakan pemain biasa. Sementara itu, *navigation mesh* mencakup seluruh titik pada area yang dicakupnya. Akibatnya, banyak kemungkinan gerakan yang dapat dilakukan untuk berpindah dari suatu area poligon ke area tetangganya sehingga dapat dipilih lintasan sedemikian sehingga jarak yang dilalui minimum dan NPC dapat berbelok dengan halus.



**Gambar 11.** Perbandingan gerakan NPC pada *waypoint graph* (kiri) dengan *navigation mesh* (kanan) pada peta yang sama  
(Sumber :

<https://docs.unrealengine.com/udk/Three/NavigationMeshReference.html> diakses pada 7 Desember 2016)

Keunggulan lain *navigation mesh* dibandingkan *waypoint graph* adalah NPC mampu berimprovisasi dengan gerakannya jika terjadi gangguan pada lintasan yang telah ditentukan semula<sup>[7]</sup>. Gangguan dapat disebabkan oleh adanya benda kecil yang menghalangi lintasan atau pertemuan dengan objek yang berbahaya. Kemampuan tersebut juga disebabkan oleh banyaknya titik yang dicakup setiap poligon. NPC dapat mengitari objek tersebut untuk melanjutkan lintasan tanpa keluar dari poligon dan tanpa merusak lintasan antar poligon yang telah dicari sebelumnya. Pada *waypoint graph*, adanya gangguan tersebut menandakan bahwa lintasan buntu dan

tidak mungkin dilalui. Jika ingin melanjutkan rute ke tujuan, gangguan harus dihilangkan atau mencari ulang lintasan tanpa memperhitungkan sisi penghubung yang terganggu.

#### IV. PENERAPAN

Dalam permainan Counter-Strike, pengguna tidak hanya dapat melawan sesama pengguna. Ada jenis NPC yang mampu bertarung seperti pemain biasa, yaitu bot. Bot dapat dijadikan musuh yang berbeda tim atau teman setim. Tingkat kesulitan bot bervariasi mulai dari sangat mudah hingga sangat sulit. Bot dengan tingkat kesulitan lebih tinggi memiliki akurasi tembakan yang lebih baik dan respon yang lebih cepat.

Untuk membantu bot menentukan gerakan dengan benar dalam peta, diperlukan *navigation mesh*. Counter-Strike menyediakan fitur untuk membuat *navigation mesh* secara otomatis dengan tujuan mempermudah para pembuat peta dari kalangan komunitas<sup>[8]</sup>. Dengan fitur tersebut, peta yang awalnya tidak memiliki *navigation mesh* tetap dapat dimainkan oleh bot setelah proses pembangunan otomatis *navigation mesh* selesai. Waktu yang diperlukan bervariasi bergantung pada ukuran dan kerumitan peta. Proses yang dilakukan dalam algoritma fitur tersebut antara lain<sup>[8]</sup>:

- Mencari seluruh area yang dapat dilalui dengan memulai dari *spawn point*, kemudian mencari titik-titik di sekitarnya yang dapat dilalui.
- Menemukan pola dari persebaran titik yang dapat dilalui dan membentuk poligon penyusun *navigation mesh*. Dalam Counter-Strike, poligon memiliki bentuk persegi atau persegi panjang.
- Memperhitungkan faktor-faktor yang mungkin memengaruhi perilaku bot di area tersebut.

Pembuat peta tetap disarankan untuk membuat *navigation mesh* secara manual, terutama jika peta yang dibuat cukup kompleks.

Pengguna dapat melihat *navigation mesh* yang telah terbentuk pada suatu peta dengan cara memasukkan perintah 'nav\_edit 1' pada konsol setelah memasukkan perintah 'sv\_cheats 1'. Untuk mengetahui informasi dan tetangga salah satu poligon, pengguna harus mengarahkan pandangannya ke poligon yang bersangkutan. Selain melihat, pengguna juga dapat mengaturnya.

Beberapa poligon memiliki atribut khusus yang berfungsi untuk mengatur gerakan bot saat melewati area tersebut. Atribut-atribut tertentu memberikan tampilan yang unik pada poligon pada mode nav\_edit. Atribut yang sering dijumpai antara lain:

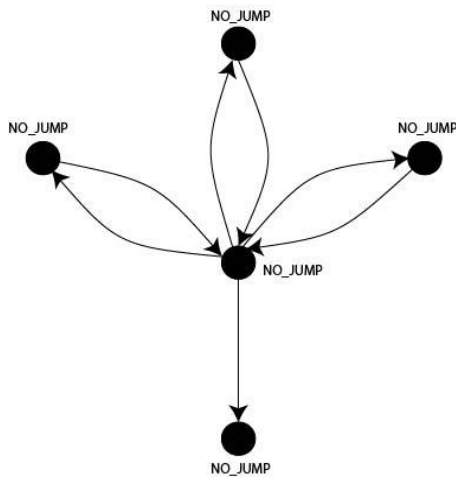
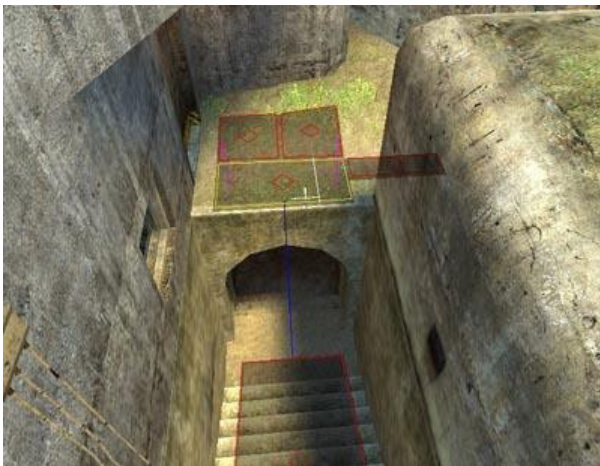
- JUMP  
Ditandai dengan dua garis diagonal berwarna hijau. Bot harus melompat saat melewatinya. Biasanya dijumpai di antara dua permukaan yang memiliki perbedaan ketinggian yang tidak terlalu tinggi.
- NO\_JUMP  
Ditandai dengan belah ketupat kecil di tengah poligon. Bot tidak diperbolehkan untuk melompat saat melewatinya.

- CROUCH

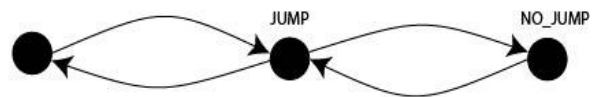
Ditandai dengan satu garis diagonal berwarna biru. Bot harus melalui poligon ini sambil berjongkok. Poligon ini dapat ditemukan pada area dengan objek dengan langit-langit rendah.

Terdapat banyak atribut lainnya. Suatu poligon dapat memiliki lebih dari satu atribut atau tidak memiliki atribut sama sekali

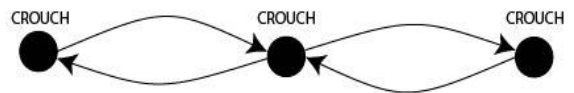
Jenis graf yang paling tepat untuk merepresentasikan *navigation mesh* dalam Counter-Strike adalah graf berarah. Meskipun dua buah poligon bersinggungan, bot belum tentu dapat bergerak bolak-balik. Contohnya adalah hubungan dua buah poligon yang berdekatan namun memiliki perbedaan ketinggian tertentu sehingga bot dapat terjun dari poligon yang terletak di atas ke poligon di bawahnya, tetapi tidak dapat melompat dari poligon bawah ke poligon atas. Hubungan satu arah tersebut direpresentasikan dengan garis berwarna biru tua yang menghubungkan kedua poligon. Garis tersebut hanya terlihat saat pemain mengarahkan pandangan ke poligon asal dan tidak tampak saat melihat poligon tujuan. Sebaliknya, jika bot mampu bergerak bolak-balik antar poligon yang bertetangga, hubungan tersebut tampak sebagai garis biru muda.



**Gambar 12.** Contoh sebagian *navigation mesh* dalam salah satu peta di Counter-Strike<sup>[8]</sup> dan graf yang merepresentasikannya



**Gambar 13.** Contoh sebagian *navigation mesh* dalam salah satu peta di Counter-Strike<sup>[8]</sup> dan graf yang merepresentasikannya



**Gambar 14.** Contoh sebagian *navigation mesh* dalam salah satu peta di Counter-Strike<sup>[8]</sup> dan graf yang merepresentasikannya

## V. KESIMPULAN

Salah satu penerapan graf dalam permainan video adalah *navigation mesh*. Dengan *navigation mesh*, dapat dicari sebuah lintasan untuk perpindahan NPC sedemikian sehingga lintasan yang dilalui minimum, memiliki pergantian arah yang halus, dan fleksibel. Saat menemui

gangguan, NPC mampu berimprovisasi dengan gerakannya tanpa merusak lintasan antar poligon komponen *navigation mesh*. Dalam permainan Counter-Strike, *navigation mesh* juga mampu mengontrol gerakan NPC selain mengatur arah gerakannya. Dengan demikian, NPC dapat bergerak dengan natural seperti sedang dikontrol oleh manusia.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa karena atas anugerah-Nya, pembuatan makalah ini berjalan dengan lancar dan dapat terselesaikan. Ucapan terima kasih juga penulis sampaikan kepada Dr. Ir. Rinaldi Munir, M.T. selaku dosen mata kuliah matematika diskrit. Penulis juga berterima kasih kepada orang tua, keluarga, dan teman-teman yang telah memberikan dukungan. Semoga makalah ini bermanfaat.

## REFERENSI

- [1] *Counter-Strike: Global-Offensive* (Video game). Valve, 2012.
- [2] Rinaldi Munir, *Diktat Kuliah IF2120: Matematika Diskrit*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.
- [3] Kenneth H. Rosen, *Discrete Mathematics and Its Application*. 7th Edition. New York: McGraw-Hill, 2012.
- [4] <http://blog.counter-strike.net/index.php/history> (diakses pada 3 Desember 2016).
- [5] <http://lambdageneration.com/discussion/counter-strike/a-history-of-counter-strike> (diakses pada 3 Desember 2016).
- [6] <http://www.giantbomb.com/source-engine/3015-751> (diakses pada 3 Desember 2016).
- [7] [https://www.cs.rit.edu/~jdb/tmp/making\\_of\\_official.pdf](https://www.cs.rit.edu/~jdb/tmp/making_of_official.pdf) (diakses pada 3 Desember 2016).
- [8] [https://developer.valvesoftware.com/wiki/Navigation\\_Meshes](https://developer.valvesoftware.com/wiki/Navigation_Meshes) (diakses pada 7 Desember 2016).
- [9] [www.cs.uu.nl/docs/vakken/mpp/other/path\\_planning\\_fails.pdf](http://www.cs.uu.nl/docs/vakken/mpp/other/path_planning_fails.pdf) (diakses pada 7 Desember 2016).
- [10] <http://mgrenier.me/2011/06/pathfinding-concept-the-basics> (diakses pada 7 Desember 2016).
- [11] <https://developer.valvesoftware.com/wiki/Bot> (diakses pada 8 Desember 2016).

## 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, 8 Desember 2016



Arno Alexander - 13515141