

Aplikasi Pohon Keputusan untuk Menentukan Perilaku AI dalam Assassin's Creed IV: Black Flag

Timothy Pratama - 13512032
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512032@stei.itb.ac.id

Makalah ini berisi tentang aplikasi dari pohon keputusan untuk menentukan bagaimana sikap dari Artificial Intelligence (AI) pada permainan Assassin's Creed IV: Black Flag. Perilaku AI di dalam permainan ini dipengaruhi oleh banyak kondisi yang diberikan oleh pemain, sehingga AI memiliki perilaku seperti manusia di dunia nyata. Di dalam makalah ini, saya akan membahas mengenai Artificial Intelligence, Teori Graf, Teori Pohon, dan Aplikasi Pohon Keputusan untuk menentukan bagaimana cara AI mengambil keputusan pada Assassin's Creed IV.

Assassin's Creed IV: Black Flag, Artificial Intelligence, Graf, Pohon

I. INTRODUCTION



Gambar 1. Assassin's Creed IV: Black Flag
(Sumber : <http://www.gamerheadlines.com/wp-content/uploads/2013/11/Assassins-Creed-4.jpg> diakses pada 7 Desember 2013 pukul 17.35 WIB)

Assassin's Creed IV: Black Flag adalah sebuah game yang memiliki *genre* open world action adventure. Permainan ini pertama kali dirilis pada tanggal 29 Oktober 2013 untuk Xbox 360 dan PlayStation 3. Baru kemudian pada bulan November 2013 versi untuk windows dirilis.

Pada permainan ini, para pemain dibawa kembali ke tahun 1715 di daerah Caribbean ketika pada saat itu para bajak laut menguasai daratan dan lautan, serta mereka juga mendirikan sebuah republik yang bebas hukum, salah satunya adalah Nassau. Para bajak laut ini menghambat perdagangan internasional karena mereka sering melakukan pembajakan dan merampas semua harta dari kapal yang melewati daerah Caribbean. Tindakan para

bajak laut ini mengancam struktur kekuasaan Eropa pada saat itu^[1].

Tokoh utama dari permainan ini adalah Edward Kenway, yaitu seorang bajak laut yang ditakuti dengan kemampuan bertarung yang handal dan dia juga mendapatkan hormat dari Blackbeard, yaitu seorang bajak laut yang melegenda^[1]. Seiring dengan berjalannya waktu, Edward pun akan menemukan konspirasi yang dilakukan oleh para Templar untuk menemukan Observatory dengan cara mencari seseorang yang bernama Sage. Pada akhirnya, Edward berhasil menemukan Observatory tersebut beserta dengan artifaknya. Namun, artifaknya jatuh ke tangan yang salah. Artifak tersebut memiliki kekuatan mistis yang dapat digunakan untuk menguasai dunia. Akhirnya, Edward bergabung dengan para Assassins dan berhasil merebut artifak itu kembali dan mengembalikannya ke Observatory.

II. ASSASSIN'S CREED IV: BLACK FLAG



Gambar 2. Tampilan gameplay Assassin's Creed
(Sumber : <http://wimages.vr-zone.net/2013/11/AC4-Bounty.jpg> diakses pada 7 Desember 2013 pukul 17.35 WIB)

Assassin's Creed IV: Black Flag memiliki tampilan seperti pada Gambar 2.

- Di pojok kiri atas terdapat informasi mengenai *health point* dari Edward, beserta dengan statusnya. Jika logo kepala berwarna putih, berarti Edward berada dalam kondisi *incognito*. Jika kuning, berarti ada musuh yang sedang mencari Edward. Dan jika merah, berarti musuh telah mengetahui lokasi Edward dan

akan terjadi konflik antara Edward dengan musuh tersebut. Selain itu, pada bagian kiri atas juga terdapat informasi mengenai misi yang sedang berlangsung, baik itu *main quest* maupun *side quest*.

- Pada bagian kiri bawah, terdapat sebuah *minimap*. *Minimap* berfungsi untuk menampilkan informasi penting yang ada di sekitar Edward, seperti toko, *restricted area*, musuh, teman, lokasi *main quest* dan *side quest*, dan tempat untuk bersembunyi saat terjadi *conflict*. Di sebelah kanan *minimap* juga terdapat informasi mengenai jumlah uang yang dimiliki oleh Edward dalam mata uang Reales.
- Pada bagian kanan atas, terdapat informasi mengenai *action* yang dapat dilakukan oleh Edward dengan menekan tombol tertentu.
- Pada bagian kanan bawah, terdapat informasi mengenai peralatan yang sedang dipakai oleh Edward.

Para pemain dapat melakukan banyak hal di dalam *game* ini. Pemain bisa memilih untuk melakukan eksplorasi ke pulau-pulau dan *shipwrecks* untuk mendapatkan barang-barang yang berharga dan uang. Pemain juga dapat berburu untuk mengumpulkan bahan untuk *crafting* dan merakit peralatan-peralatan yang dapat meningkatkan kemampuan Edward. Pemain juga dapat memilih untuk melakukan *main quest* untuk melanjutkan *storyline* dan *side quest* untuk mendapatkan barang-barang berharga maupun uang. Pemain juga dapat melakukan berbagai aktivitas di laut, seperti merebut sebuah *fort* maupun membajak sebuah kapal.

Selama melakukan eksplorasi atau misi, terdapat dua jenis daerah. Daerah yang ditandai dengan warna merah berarti *restricted area*, sedangkan daerah yang tidak ditandai berarti daerah netral. Pada saat Edward berada di *restricted area*, jika Edward terlihat oleh musuh, maka *suspicion meter* akan bertambah. Apabila *suspicion meter* telah penuh, maka musuh akan menyerang Edward dan memanggil teman-temannya. Pada daerah yang tidak terlarang, Edward dapat berjalan-jalan bebas selama tidak melakukan hal-hal yang mencurigakan atau menyerang tentara musuh secara langsung. Jika pemain menimbulkan masalah, maka tentara musuh pun akan datang dan menyerang Edward. Begitu juga pada saat Edward melakukan *main quest* maupun *sidequest*. AI selalu membuat keputusan berdasarkan kondisi dari Edward.

III. TEORI DASAR

A. Artificial Intelligence

Artificial Intelligence, atau biasa disingkat sebagai AI, dapat didefinisikan sebagai sebuah sistem yang berpikir seperti manusia, bertindak seperti manusia, berpikir dengan rasional, dan bertindak secara rasional^[2]. Pada Tahun 1950, Alan Turing menciptakan The Turing Test untuk mendefinisikan bahwa *intelligent behavior* adalah kemampuan untuk mencapai performa yang memiliki level yang sama dengan manusia dalam semua pekerjaan yang bersifat kognitif. Jika orang yang berinteraksi dengan komputer tidak dapat membedakan apakah yang menjawab adalah sebuah mesin atau orang, maka

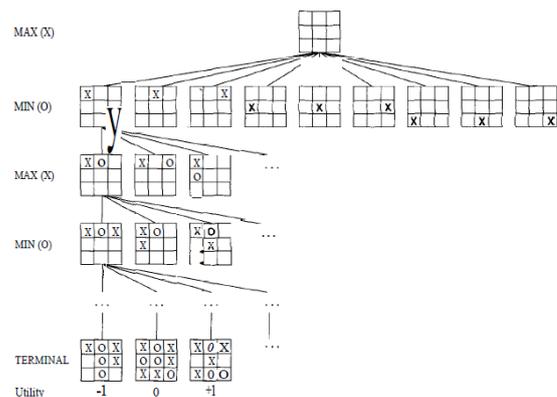
komputer tersebut dinyatakan lulus test.

Ada beberapa strategi yang dapat digunakan untuk menemukan sebuah solusi. Setiap strategi dapat dievaluasi berdasarkan 4 kriteria, yaitu^[2]

- *completeness* : apakah strategi dapat menemukan sebuah solusi jika memang ada sebuah solusi?
- *time-complexity* : waktu yang dibutuhkan untuk menemukan sebuah solusi.
- *space-complexity* : memori yang dibutuhkan untuk melakukan pencarian.
- *optimality* : apakah strategi dapat menemukan solusi yang terbaik dari solusi-solusi yang ada?

Strategi yang ada dalam melakukan pencarian adalah *breadth-first search*, *uniform cost search*, *depth-first search*, *depth-limited search*, *iterative deepening search*, dan *bidirectional search*.^[2]

AI dalam *game* pada umumnya bertujuan untuk menciptakan sebuah *hostile agent* yang bertidak seperti manusia^[2]. Dalam *game* yang melibatkan dua pihak, adanya musuh akan memunculkan aspek ketidakpastian, karena seseorang tidak pernah tahu apa yang akan dilakukan oleh musuh tersebut. Sehingga yang membuat *game* berbeda adalah pada umumnya *game* terlalu sulit untuk dipecahkan, karena AI dalam *game* harus memperhitungkan berbagai kemungkinan yang ada. Misalkan pada *game* Tic-Tac-Toe, maka semua kemungkinan langkah yang mungkin dilakukan oleh musuh dapat digambarkan dalam sebuah *Search Tree*.



Gambar 3. Search Tree pada game Tic-Tac-Toe (Sumber : referensi^[2])

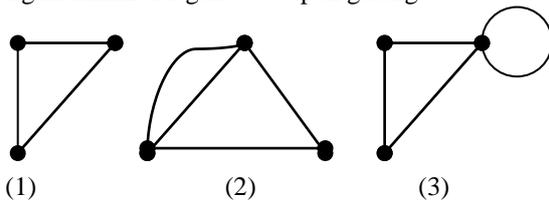
Seperti pada gambar 3, dengan menggunakan sebuah *Search Tree* maka AI dapat menentukan langkah apa yang harus diambil berdasarkan langkah yang dilakukan oleh musuh. Begitu juga dengan *game* lainnya, dengan menggunakan *search tree* maka AI dapat menentukan langkah yang harus diambil terhadap langkah yang dilakukan oleh musuh. Selain menggunakan *search tree*, salah satu cara lain untuk memodelkan bagaimana cara AI berpikir adalah dengan menggunakan pohon keputusan. AI akan membuat keputusan berdasarkan kondisi yang ada.

B. Graf

Graf didefinisikan sebagai pasangan himpunan (V,E),

dimana V adalah himpunan tidak kosong dari simpul-simpul dan E adalah himpunan dari sisi yang menghubungkan sepasang simpul^[3]. Graf dapat digunakan untuk merepresentasikan berbagai permasalahan yang ada, misalnya graf dapat digunakan untuk merepresentasikan hubungan antara satu kota dengan kota yang lain dalam bentuk peta.

Berdasarkan ada atau tidaknya gelang atau sisi-ganda, maka graf dapat dikelompokkan menjadi 2 jenis, yaitu graf sederhana dan graf tak-sederhana^[3]. Sebuah graf dikatakan memiliki sisi ganda jika ada satu pasang simpul yang dihubungkan oleh lebih dari satu sisi. Sedangkan graf dikatakan memiliki gelang jika ada sebuah sisi yang menghubungkan sebuah simpul dengan dirinya sendiri. Graf sederhana adalah graf yang tidak memiliki gelang maupun sisi ganda. Sedangkan graf tak-sederhana adalah graf yang memiliki sisi ganda ataupun gelang.



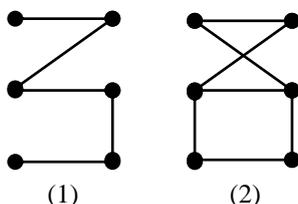
Gambar 4. (1) graf sederhana, (2) graf tak-sederhana dengan sisi ganda, (3) graf tak-sederhana dengan gelang

Sedangkan berdasarkan ada atau tidaknya arah pada sisi graf, maka graf dapat dibedakan menjadi graf berarah dan graf tak-berarah^[3]. Graf berarah adalah graf yang sisinya memiliki arah, ditunjukkan dengan adanya anak panah pada sisi graf. Misalkan ada dua buah simpul. Kedua simpul tersebut dihubungkan oleh sebuah sisi yang berasal dari simpul pertama menuju simpul kedua. Graf tersebut hanya terhubung dari simpul kesatu ke simpul kedua, tetapi tidak sebaliknya. Dalam graf berarah, sisi biasa disebut juga sebagai busur. Sedangkan graf tak-berarah adalah graf yang sisinya tidak memiliki arah. Dua buah simpul yang dihubungkan dengan sebuah sisi tetap dapat dikatakan terhubung tanpa memerhatikan arah dari sisinya.

Graf dapat diterapkan untuk berbagai permasalahan yang ada. Salah satu konsep dalam teori graf yang penting adalah pohon. Pohon akan dibahas dalam sub-bab berikutnya.

C. Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit^[3]. Artinya, pohon adalah sebuah graf dimana di dalam graf itu semua simpul harus terhubung satu dengan yang lain dan tidak boleh ada sirkuit di dalam graf tersebut.



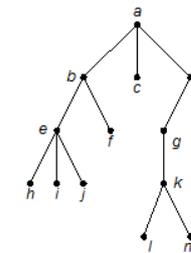
Gambar 5. (1) adalah pohon, (2) bukan pohon

Beberapa pohon dapat membentuk hutan. Hutan adalah kumpulan dari pohon-pohon yang saling lepas. Dengan kata lain, hutan adalah graf tidak terhubung yang tidak mengandung sirkuit^[3].

Misalkan G adalah sebuah pohon, maka G memiliki sifat-sifat seperti^[3]:

1. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
2. G terhubung dan memiliki $e = v - 1$ buah sisi.
3. G tidak mengandung sirkuit.
4. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
5. G terhubung dan semua sisinya adalah jembatan.

Di dalam pohon berakar terdapat beberapa terminologi yang cukup penting, yaitu^[3]:



Gambar 6. Pohon berakar (sumber : referensi^[3])

1. Anak dan Orangtua
Simpul a dikatakan anak dari simpul b jika ada sisi yang menghubungkan a dan b dan aras a lebih besar dari b . Pada gambar 6, b, c, d adalah anak dari a , sedangkan a adalah orangtua dari b, c, d .
2. Lintasan
Lintasan dari simpul a ke simpul b adalah simpul-simpul yang dilintasi untuk sampai ke b dari a . Pada gambar 6, Lintasan dari a ke f adalah a, b, f . Panjang lintasan a ke f adalah 2.
3. Saudara kandung
Saudara kandung adalah simpul yang memiliki orangtua yang sama. Pada gambar 6, j adalah saudara kandung dari h , tetapi i bukan saudara kandung dari h .
4. Keturunan dan Leluhur
Jika terdapat lintasan dari a ke b di dalam pohon, maka a adalah leluhur dari b dan b adalah keturunan dari a . Pada gambar 6, a adalah leluhur dari e , sedangkan e adalah keturunan dari a .
5. Derajat
Pada pohon berakar, derajat adalah jumlah anak yang dimiliki oleh sebuah simpul. Misalkan pada gambar 6, simpul a memiliki derajat sebesar 3.
6. Daun dan Simpul Dalam
Daun adalah simpul yang tidak mempunyai anak. Sedangkan simpul dalam adalah simpul yang memiliki anak. Pada gambar 6, yang disebut dengan daun adalah simpul h, i, j, l, m . Sedangkan yang disebut dengan anak dalam adalah simpul b .
7. Aras
Akar mempunyai aras = 0, sedangkan simpul lainnya mempunyai aras = panjang lintasan dari akar ke simpul tersebut. Contohnya, pada gambar 6, simpul b, c, d memiliki aras = 1.

8. Tinggi atau Kedalaman

Tinggi atau kedalaman adalah aras maksimum dari sebuah pohon. Contohnya pada gambar 6, pohon tersebut memiliki kedalaman sebesar 4.

Pohon biner adalah pohon yang setiap simpul cabangnya paling banyak memiliki 2 buah anak, yaitu anak kiri dan anak kanan^[3]. Salah satu terapan dari pohon biner adalah pohon keputusan. Pohon keputusan digunakan untuk memodelkan sebuah masalah dengan berbagai keputusan yang harus diambil untuk mendapatkan solusinya. Di dalam pohon keputusan, simpul dalam menyatakan keputusan yang diambil, dan setiap daun menyatakan solusi yang diambil. Pohon keputusan dapat digunakan untuk menentukan sikap dari AI, contohnya adalah pohon keputusan untuk menentukan langkah dalam permainan Tic-Tac-Toe pada gambar 3. Solusi yang diambil berdasarkan langkah yang dilakukan oleh pemain.

IV. APLIKASI POHON KEPUTUSAN DALAM ASSASSIN'S CREED IV

Di dalam Assassin's Creed IV, Edward bisa bertemu dengan beberapa jenis musuh, yaitu tentara Spanyol, tentara Inggris atau bajak laut sendiri. Setiap musuh terdiri dari beberapa kelas, mulai dari tentara biasa, *brute*, *captain*, dan *rifleman*. Selain itu, tempat bertemu dengan musuh tersebut juga ada dua macam, pertama adalah *restricted area* dan yang kedua adalah daerah netral. *Restricted area* biasanya ditandai dengan warna merah pada *minimap*. *Restricted area* bisa berupa sebuah kantor gubernur, benteng, seseorang dengan pangkat yang tinggi (biasanya ada pada saat melakukan *main quest*), daerah di sekitar gudang tempat menyimpan barang untuk perdagangan, dan sebagainya. Sedangkan daerah yang netral adalah daerah yang bebas dimasuki oleh siapa saja, seperti jalan pada kota-kota besar, pulau, dan sebagainya.

Jika Edward bertemu dengan musuh pada daerah netral, maka pada umumnya AI akan menganggap Edward sama seperti orang biasa. AI tidak akan mengambil tindakan apapun dan menganggap Edward seperti tidak ada. Tetapi jika Edward melakukan tindakan yang ilegal, seperti memanjat bangunan atau menabrak musuh, maka pada musuh tersebut pun akan ditampilkan sebuah *suspicion meter* yang akan bertambah selama tindakan tersebut masih dilakukan atau Edward tidak menghilang dari jarak pandang musuh tersebut. *Suspicion meter* juga akan bertambah jika musuh tersebut menemukan adanya mayat. Jika *suspicion meter* sudah penuh, maka AI kemudian akan mencari Edward dan menyerangnya. Jika aksi ilegal yang dilakukan adalah menyerang atau membunuh musuh tersebut secara terang-terangan, maka AI pun akan langsung menyerang balik. Tetapi jika musuh diserang secara diam-diam, seperti menggunakan *hidden-blade*, dari tempat sembunyi, ataupun dengan menggunakan *tools* lainnya, maka AI tidak akan menyerang.

Sedangkan untuk kasus tempat bertemunya di *restricted area*, maka jika Edward terlihat oleh musuh, *suspicion meter* pun akan bertambah, walaupun Edward tidak melakukan tindakan apapun. Jika *suspicion meter* sudah

penuh, maka AI akan langsung menyerang Edward.

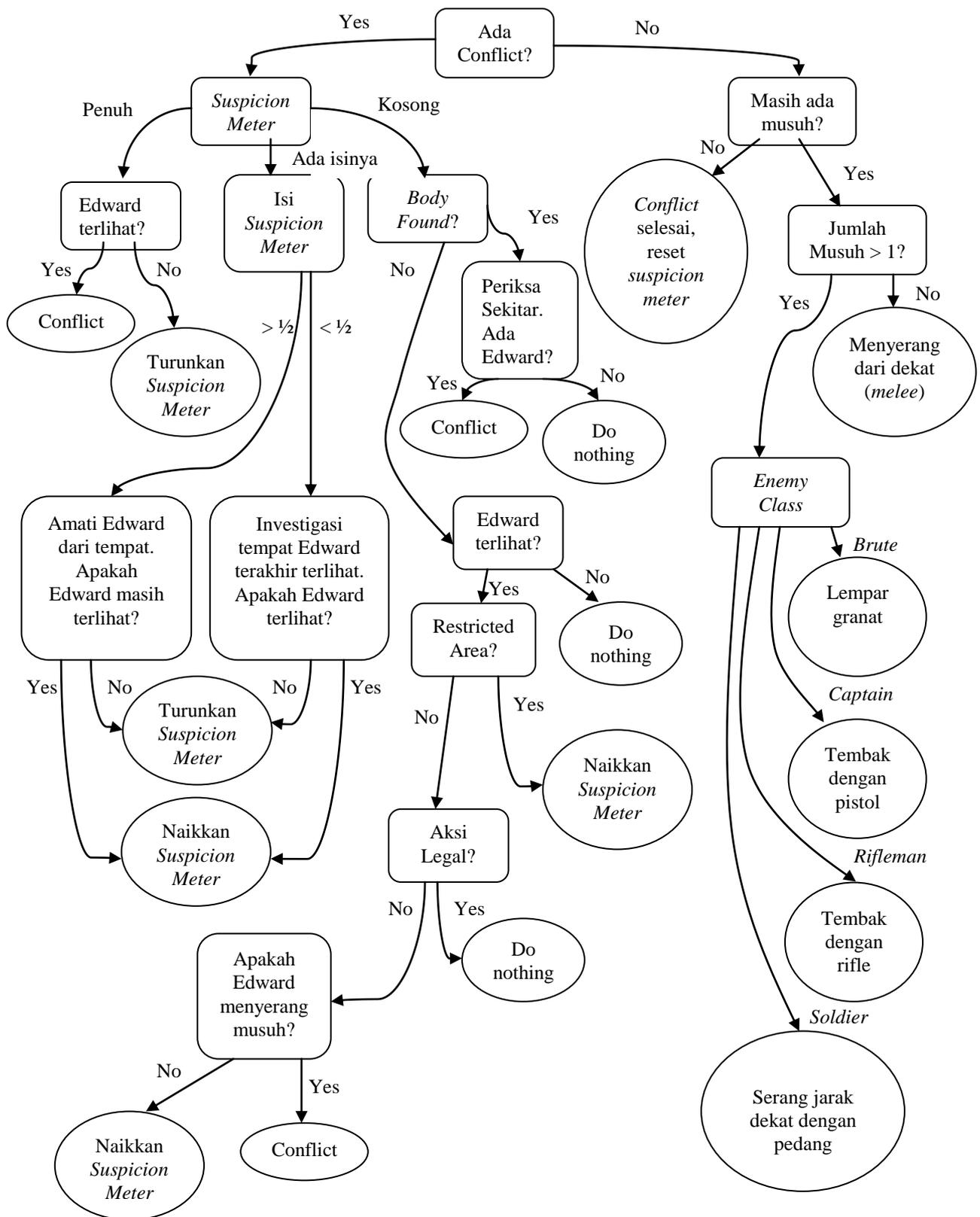
Kondisi pada saat musuh sedang mencari Edward atau pada saat musuh sedang menyerang Edward disebut sebagai *conflict*. Pada saat terjadi *conflict*, jika disekitar area *conflict* terdapat tentara musuh, maka musuh pun akan berdatangan dan menyerang Edward. Jika pada saat *conflict* Edward atau musuh menembakkan senjata, maka musuh pada jarak yang cukup jauh pun akan datang karena 'mendengar' suara tembakan tersebut.

Conflict akan berakhir dengan beberapa cara. Pertama, semua musuh yang sedang mencari Edward tidak dapat menemukan Edward. Kedua, semua musuh yang sedang mencari atau menyerang Edward berhasil dikalahkan oleh Edward.

Keputusan yang diambil oleh AI dalam Assassin's Creed IV ini sangat bergantung terhadap kondisi Edward. Pohon keputusan dapat digunakan untuk memodelkan persoalan ini, yaitu keputusan apa yang diambil dan apa solusinya. Dengan adanya pohon keputusan, maka AI dapat mengetahui apa yang harus dilakukan ketika Edward melakukan sesuatu, karena AI memiliki data mengenai aksi apa yang dilakukan oleh Edward dan respon apa yang harus diberikan berkaitan dengan aksi Edward tersebut. Setiap simpul dalam melambangkan pertimbangan yang dipertimbangkan oleh AI, kemudian daun melambangkan solusi atau langkah yang harus diambil oleh AI. Setelah mendapatkan solusi, maka berikutnya AI akan kembali ke awal lagi untuk menentukan sikap apa yang harus dilakukan. Misalnya, ketika AI melihat Edward berada di *restricted area*, maka AI akan menaikkan *suspicion meter*. Ketika *suspicion meter* masih di bawah setengah, maka AI hanya akan mengamati Edward dari tempat dengan terus menaikkan *suspicion meter* selama Edward masih terlihat. Jika *suspicion meter* di atas setengah dan Edward tidak terlihat, maka AI akan menghampiri tempat terakhir Edward terlihat. Jika Edward ditemukan, maka *suspicion meter* akan penuh dan *conflict* pun terjadi. Pada saat *conflict* berakhir, maka AI akan memulai dari awal lagi, yaitu memberikan respon sesuai dengan keadaan Edward pada saat *suspicion meter* masih kosong. Dengan menggunakan pohon keputusan, maka persoalan yang tampaknya rumit pun dapat dibuat lebih sederhana, sehingga *programmer* dapat membuat program dengan lebih jelas.

Setelah memainkan Assassin's Creed IV cukup lama, maka saya mencoba untuk membuat pohon keputusan yang digunakan oleh AI pada Assassin's Creed IV. Pohon keputusan ini ada di halaman berikutnya.

Pada gambar 7 di bawah ini, solusi yang diambil oleh AI dilambangkan dengan lingkaran, sedangkan keputusan yang diambil oleh AI dilambangkan dengan persegi. Jika solusi yang diambil do nothing, berarti AI akan tetap melakukan tugas yang sedang dilakukan oleh musuh pada saat itu, misalnya berpatroli keliling kota, mengobrol dengan sesama tentara, dan sebagainya.



Gambar 7. Pohon keputusan AI dalam Assassin's Creed IV

V. KESIMPULAN

Pohon dapat digunakan untuk memodelkan banyak persoalan, salah satunya adalah persoalan AI dalam game. Dengan menggunakan pohon keputusan, maka permasalahan yang berkaitan dengan membuat keputusan untuk menemukan solusi dapat digambarkan dengan lebih sederhana. Sehingga pada saat melakukan pemrograman untuk AI, pemrogram akan lebih mudah karena mengetahui bagaimana sikap dari AI dan apa saja yang harus dipertimbangkan oleh AI.

REFERENCES

- [1] <http://assassinscreed.ubi.com/en-GB/home/index.aspx>
Diakses pada tanggal 7 Desember 2013 pukul 10.45 WIB.
- [2] Russell, Stuart J., Norvig, Peter. 1995. Artificial Intelligence : A Modern Approach. New Jersey : Prentice-Hall, Inc.
- [3] Munir, Rinaldi. 2003. Matematika Diskrit Edisi Kedua. Bandung: Penerbit Informatika.

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, 14 Desember 2013



Timothy Pratama
13512032