

# Kecerdasan Buatan dengan Metode ID3 Finite State Machine dalam Turn-Based Tactics Game

Adam P. Sulaiman, Liliana, Leo Willyanto Santoso.

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jln. Siwalankerto 121-131 Surabaya 60236 Telp.(031)-2983455, Fax.(031)-8417658

E-mail: M26415024@john.petra.ac.id, lilian@petra.ac.id, leow@petra.ac.id

## ABSTRAK

Seiring dengan perkembangan dunia *video game* maka bertambah pula pemain yang bergantung kepada *video game*. Salah satu cara untuk meningkatkan immersi pemain dalam sebuah permainan adalah melalui AI (Kecerdasan buatan). AI singkat cerita adalah sebuah kemampuan di dalam sebuah mesin yang dapat memberikan kemampuan kepada mesin untuk dapat “berpikir” sehingga dapat melakukan sesuatu kegiatan tanpa input langsung dari manusia. Dan AI telah berkembang secara pesat dan konstan dari masa ditemukan pertama kali.

Namun kelemahan terbesar dari AI adalah apabila penerapan struktur didalam pembuatan permainan terlalu sederhana, maka apabila pola AI telah diketahui, maka pola tersebut dapat dimanfaatkan oleh pemain sehingga mengakibatkan hilangnya immersi dikarenakan permainan dimanfaatkan untuk menjadi lebih mudah atau dimainkan dengan cara yang tidak seharusnya. Untuk menyelesaikan masalah tersebut, maka dibutuhkan sebuah AI fleksibel yang dapat mengganti pola bermain mereka selama permainan berjalan. Dengan menggunakan ID3 decision tree, AI dapat memutuskan untuk menggunakan berbagai variasi cara berpikir yang dapat digunakan di dalam sebuah Turn-Based Strategy Game. Dan diharapkan dengan sedikit perkembangan, AI tersebut dapat digunakan untuk jenis permainan yang berbeda.

Hasil pengujian membuktikan bahwa ID3 memberikan pengaruh kepada kemampuan AI dalam mengambil keputusan, dan setelah tes percobaan program untuk jenis permainan Turn-Based Game, AI jenis adaptif yang menggunakan ID3 menunjukkan keunggulan daripada AI statik dan AI acak karena AI ini memilih nilai secara dinamik menggunakan data permainan menunjukkan hasil rasio dengan pemain 1:2:0, AI statik yang selalu memilih hasil terbaik dengan rasio 1:1:1 dan AI acak selalu di dalam kondisi yang merugikan dengan rasio 0:0:3. Pengaruh dari ID3 tidak ada pada sesi awal permainan dan akan meningkat seiring dengan meningkatnya nilai nilai pada tabel data yang digunakan.

**Kata Kunci:** Kecerdasan Buatan, ID3, *Finite-State Machine*, Permainan strategi, Unity, FSM, 2D, *Decision tree*, *Video Game*, *Turn-Based Strategy*, *Turn-Based Tactics*

## ABSTRACT

*As the world of video game grown and advances there are even more those who likes and play them as well. One way to increase immersion within video games is by implementing AI (Artificial Intelligence). AI is an ability within a program or machine that can grant said machine/program a power of “thought” so that they can do a command without unnecessary human inputs. And AI had grown constantly significant since the day it was conceived.*

*However one of AI's greatest weakness is if implemented too simplistic within it's execution, then when someone manage to decode the AI's pattern, then said player can use this knowledge for their advantage. And as a consequence, costed the game immersion since the game is played not the way it was intended to or become way too easy. To solve said problem, requires a flexible AI that can adapt to the gameplay. Using ID3 decision tree, AI are expected to be able to using the variable within a Turn-Based Strategy Game to use different options according to the data it earn. And hopefully with little improvements, this method can be used to differing video game genres as well.*

*The test result proven that while ID3 does give some influence for the AI decision making, during the test run for Turn-Based Game shown that the Adaptive AI that always chooses the best option if played the same had a WDL overalls (Win-Draw-Lose) of 1:2:0, while Adaptive AI with the ID3 formula and dynamic decision making had a more balanced results of 1:1:1 and random AI always ended at a disadvantage with 0:0:3. These tests proved that Adaptive AI provide the most challenge to the player doesn't have any influence for the early parts of the game. And that the ID3 impact were non-existent during the early game but improving as the game goes on and the data table more varied.*

**Keywords:** *Artificial Intelligence, ID3, Finite-State Machine, Strategy Game, Unity, FSM, 2D, Decision tree, Video Game, Turn-Based Strategy, Turn-Based Tactics*

## 1. PEMBUKAAN

Pada umumnya, sebagian besar permainan komputer (*Video Game*) akan menerapkan kode skrip dimana bagian permainan yang tidak dikendalikan oleh pemain diatur oleh sistem internal buatan yang disebut sebagai AI (*Artificial Intelligence/Kecerdasan Buatan*).

Ada berbagai cara untuk menerapkan kecerdasan buatan di dalam dunia digital. Salah satu penerapan yang paling umum adalah *video game*. kecerdasan buatan telah menjadi sebuah bagian yang tidak dapat dipisah dikarenakan esensi penggunaannya untuk permainan dapat berjalan. Sebelumnya *video game* hanya dapat dimainkan secara sederhana dan harus dimainkan oleh setidaknya dua pemain.

Tujuan utama dari skripsi ini adalah mempelajari pendalaman kecerdasan buatan melalui pola *state machine* (tahapan mesin) khususnya dengan ID3 (*Iterative Dichtomiser 3*), dimana berbagai tahapan akan memilih pilihan yang sama untuk membuat pola kecerdasan buatan di dalam permainan *Turn-Based*. Hal ini juga berkaitan dengan jenis penerapan kecerdasan buatan yang tidak memiliki perubahan pada pola tahapannya (*reactive machine*) dimana pola kecerdasan buatan akan selalu sama dalam situasi

apapun dan kecerdasan buatan yang dapat berubah pola tahapannya berdasarkan tindakan pemain (*limited memory*) dalam sesi permainan tersebut (namun kembali seperti awal ketika sesi diulang) Mengapa *limited memory*? Karena dengan jenis ini, pola komputer akan berubah berdasarkan dengan data yang diberikan pemain memberikan variasi kepada permainan dan membuat suatu situasi dimana pihak musuh akan beradaptasi dengan tindakan yang dilakukan pemain sehingga pemain akan bertindak sesuai dengan adaptasi kecerdasan buatan tersebut.

Salah satu kelemahan dari kecerdasan buatan adalah pada tahap pembuatannya. Hal ini dikarenakan kecerdasan buatan tidak dapat merubah pola kerja mereka setelah dibuat dan apabila ada perubahan pada pola kecerdasan buatan yang ada, maka pola kecerdasan buatan yang lama harus dihapus untuk dijiplak dengan pola yang baru. Ini merupakan salah satu alasan mengapa perbaikan (*patch*) kepada permainan jarang menyentuh kecerdasan buatan dikarenakan resiko kerusakan kecerdasan buatan pada permainan. Walaupun permainannya sederhana atau simpel, hal ini pada umumnya juga mengakibatkan desain kecerdasan buatan yang lama juga diganti pula.

Tujuan dari Skripsi ini adalah membuat, mempelajari dan memperdalam pengetahuan, pengembangan dan pemanfaatan interaksi antara komputer dengan pemain secara efisien dalam kemampuan adaptabilitas (kemampuan untuk menyesuaikan rutinitas dalam sebuah sesi permainan) kecerdasan buatan permainan komputer melalui *finite state machine* di mana tingkat interaksi antara pemain dengan kecerdasan buatan mempengaruhi pola pikir kecerdasan buatan dan memberikan penyelesaian masalah yang berbeda dalam sebuah permainan dan mendalami implementasi dan relasi interaksi kecerdasan buatan dalam permainan sehingga dapat memahami cara pembuatan kecerdasan buatan yang *non-statis* (tidak tetap).

## 2. DASAR TEORI

### 2.1 Kecerdasan Buatan

Kecerdasan buatan (*Artificial Intelligence* atau AI) adalah sebuah studi fakultas mental dalam pembuatan mesin/perangkat lunak dengan penggunaan rumus komputer untuk memberikan perangkat kemampuan untuk memahami dan menyelesaikan sebuah aktivitas tanpa bantuan manusia. Dibuat pada abad ke 20 (tahun 1956) dan AI telah berkembang dengan sangat pesat sejak pembuatannya dan telah meningkatkan proses teknologi dan kemampuan tenaga kerja dalam berbagai hal[16].

Rumus urutan AI pada umumnya adalah kemampuan bereaksi dan menganalisa apa yang mereka tangkap dalam waktu tersebut (Intelejensi: kemampuan menangkap + menganalisa + bertindak)<sup>[2]</sup>. Hal ini membuat AI terlihat mampu untuk meniru kecerdasan yang dimiliki manusia dalam penyelesaian masalah. Kecerdasan ini mempengaruhi berbagai hal di dalam kehidupan manusia terutama dalam pelaksanaan penyusunan dan administrasi kerangka tenaga kerja manusia dengan mesin. Namun persepsi AI jauh lebih terbatas dibandingkan kecerdasan manusia meskipun AI memiliki berbagai kelebihan dari sisi konsistensi, efisiensi, kecepatan dan duplikasi. Hal ini disebabkan oleh keterbatasan AI dimana mereka tidak dapat memikirkan kemungkinan diluar keterbatasan pembuatnya[16].

Ada berbagai jenis pembelajaran di dalam kecerdasan buatan yang meliputi: Sistem pakar yang mempelajari ketidakpastian di dalam sistem, prosesor pemahaman bahasa, pemahaman suara, sistem sensor, robotik, komputer neural, pembelajaran situasi, dll.

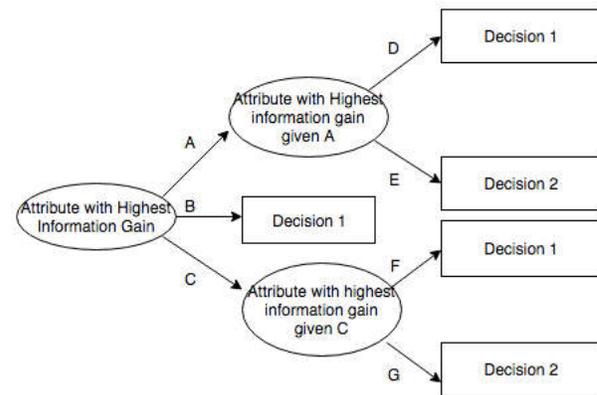
Pembelajaran tersebut dapat diaplikasikan ke berbagai penerapan seperti desain arsitek, finansial, bisnis perusahaan, industri perangkat lunak dan permainan komputer, pembacaan cuaca dst[16].

### 2.2 Video Game

*Video game* (Permainan komputer) adalah sebuah bentuk media hiburan yang menggunakan sebuah mesin konsol atau komputer dimana pengguna media tersebut bermain untuk menyelesaikan sebuah tujuan di dalam permainan tersebut. AI dan *video games* sudah menjadi hal yang tidak dapat terpisahkan dikarenakan AI merupakan sebuah bentuk yang penting dimana digunakan untuk membuat lingkungan yang tidak dimainkan oleh pemain[1].

Dalam bukunya Adams[5] menjelaskan bahwa “*a game is a type of play activity, conducted in the context of a pretend reality, in which the participant(s) try to achieve at least one arbitrary, nontrivial goal by acting in accordance with rules.*” Yang apabila diterjemahkan ke dalam bahasa indonesia menjadi “sebuah *game* adalah sebuah jenis aktivitas permainan yang dilakukan dalam konteks meniru realita dimana sebuah peran dan pemain berusaha untuk mencapai tujuan tertentu, dan bertindak sesuai dengan aturan”. Sedangkan Pressman dalam Ramadhan dan Widyani[14] berpendapat bahwa “*game is kind of software wich provides entertainment*“ yang berarti sebuah *game* adalah sebuah *software* yang menyediakan hiburan[12].

### 2.3 ID3 Decision Tree



Gambar 1. Contoh ID3 Decision Tree.

*Decision Tree* adalah sebuah struktur yang terdiri dari berbagai *nodes* dan *edges* yang memberikan tampilan menyerupai sebuah keputusan (noda keputusan) atau hasil (noda ujung) dimana akhir tersebut dipilih berdasarkan kriteria yang ada pada setiap noda. Pada dasarnya, *Decision Tree* merupakan rumus *automata* yang paling sederhana dalam pembuatannya. *Decision Tree* sangatlah berguna ketika akan menggunakan sebuah prosedur dimana terdapat lebih dari satu pilihan dan memiliki berbagai faktor, namun pilihan yang terlalu banyak akan menyebabkan kesulitan dalam pembuatannya. Noda awal atau *root node* dan *edges* (pilihan) pada umumnya merupakan opsi *Y/N* (*Fit* atau *Unfit*). Noda yang berada diantara noda awal dan noda akhir disebut sebagai noda internal/tengah dan Noda hasil disebut sebagai *leaf nodes*. [17]

ID3 (Iterative Dichotomiser 3) adalah salah satu metode decision tree yang ditemukan oleh Ross Quinlan (1983) [13]. Metode ini dapat berkembang secara otomatis (belajar) dengan menggunakan data yang sebelumnya telah disediakan oleh program, atau dengan menggunakan penambahan data yang didapatkan seiring berjalannya program tersebut. Yang membedakan ID3 dengan Decision Tree adalah penggunaan data dalam upaya mencari node akhir dimana apabila Decision Tree biasa hanya menggunakan data yang sudah disediakan sebelum *Tree* dijalankan, maka ID3 dapat memanipulasi data yang ada selama proses pemilihan berjalan. Manipulasi data ini dilakukan dengan cara merekam metode pergerakan pengguna program selama sesi berjalannya program tersebut. Dengan adanya proses ini, maka node akhir dari sebuah decision tree dapat merangkul hasil akhir yang sama walaupun berada di node yang berbeda.

Pembentukan decision tree yang optimal dari ID3 akan memerlukan sejumlah pertanyaan data yang tidak terlalu besar (sehingga mengurangi kedalaman dari *decision tree*), namun diperlukan untuk menentukan atribut pertanyaan apakah yang paling penting (yang berposisi di *root node*). Berikutnya diperlukan data berisi atribut yang ditanyakan serta atribut hasil, kumpulan dari data disebut dengan *dataset* dimana *dataset* ini akan digunakan untuk dapat melakukan proses pemilihan data.

Untuk dapat menemukan atribut apakah yang terpenting dari sejumlah atribut pertanyaan yang sudah ditentukan, diperlukan adanya proses untuk menentukan *information gain* dari setiap atribut-atribut yang tersedia. Hasil *information gain* kemudian akan dibandingkan dan diambil yang memiliki nilai terbesar. *information gain* dengan nilai terbesar menunjukkan atribut yang bersangkutan adalah atribut yang paling menentukan hasil (output) tindakan, sehingga diletakkan di bagian atas dari decision tree.[13]

Jika diinginkan, ID3 dapat menggunakan seluruh *entropy* semaksimal mungkin (tanpa membuang *entropy* yang nilainya kecil). Rumus dari *entropy*[8] adalah

$$\{Entropy(S) = - \sum p_i * \log_2(p_i) ; i = 1 \text{ to } n \} : \quad (1)$$

- (S) = Nilai yang akan dicari
- Pi = Probabilitas dari kelas (i) {rasio jumlah baris kelas i dalam kolom tujuan dan jumlah baris}
- i = indeks
- N = Jumlah kelas dari kolom target

Untuk menentukan sebuah Pendapatan Informasi (IG/*Information Gain*)[14]:

$$\{IG(S, A) = Entropy(S) - \sum ((|S_v| / |S|) * Entropy(S_v))\} : \quad (2)$$

- IG = Information Gain
- (S,A) = S adalah nilai yang dicari dan A adalah nilai yang diketahui
- Sv = Jumlah set baris dimana A adalah kolom memiliki nilai v
- |Sv| = jumlah baris di Sv dan |S| adalah jumlah baris S

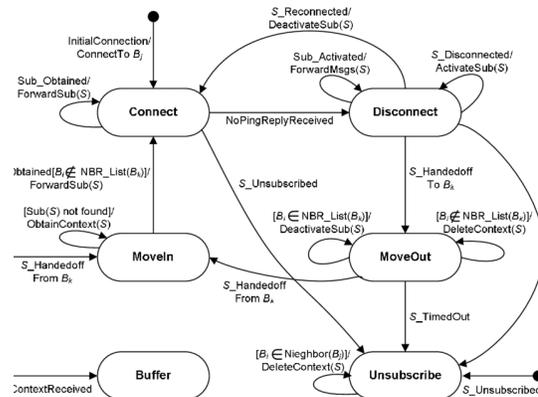
Hal ini dilakukan untuk memastikan jumlah node *output (result)* sesuai dengan jumlah hasil pada data latihan yang menghasilkan *decision tree* yang memiliki klasifikasi secara sempurna sesuai dengan data yang disediakan.

Hal-Hal penting yang butuh diperhatikan dalam pembuatan Decision Tree ID3 adalah[13]:

1. Hitung Information Gain dalam setiap node
2. Setiap baris belum tentu berada dalam sebuah kelas yang sama, pecah S menjadi beberapa subset menggunakan rumus dimana IG mendapatkan hasil tertinggi
3. Buat Decision Tree dari hasil 2 dari hasil maksimum IG
4. Apabila semua baris berada dalam kelas yang sama, buatlah node akhir dengan menggunakan nama kelas sebagai label node
5. Ulangi tahap yang 4 ke kemungkinan yang tersisa sampai semua kemungkinan sudah ditemukan atau Decision Tree sudah memiliki semua leaf nodes.

Kelemahan utama dalam *decision tree* ID3 adalah ketidakadanya kemampuan decision tree untuk melakukan pemangkasan (*pruning*) data menjadi data yang sederhana dan ID3 memiliki tingkat varian yang terlalu tinggi (mempelajari data dengan sebegitu detail) sehingga ID3 tidak dapat digunakan untuk menggunakan data dengan nilai yang berbeda tanpa membuat sebuah dataset baru (*data generalization*)[17].

## 2.4 Finite State Machine



Gambar 2. Contoh FSM

FSM (*Finite-state Automaton*) adalah sebuah desain model kalkulasi dengan bantuan *hardware* maupun *software* untuk membuat runtutan logika yang digunakan oleh program untuk mensimulasikan sebuah kegiatan atau keputusan logis dalam eksekusi aktivitas berdasarkan data yang diambil saat program dijalankan. FSM dapat diterapkan dalam berbagai mesin dan merupakan salah satu cabang Teori Automata dimana mesin dapat melakukan aktivitas yang berbeda beda berdasarkan berbagai pemicu, namun hanya satu aktivitas saja yang dapat dilakukan dalam satu waktu tertentu.[21]

Setiap FSM memiliki berbagai tahap (*state*) yang dapat dipilih dan hanya satu tahap saja yang dapat aktif per aksi sebelum FSM bertransisi ke tahap berikutnya. Perubahan tahapan ini didasarkan kepada input yang menghasilkan output untuk mensimulasikan perubahan ke tahap selanjutnya. Pada umumnya FSM memiliki sebuah tahap netral (*default state*) sebagai tahap awal dan menjadi titik kembali tahap apabila AI tidak dapat menemukan pemicu untuk dapat bertransisi ke tahap apapun.[9]

Selain itu, FSM pada umumnya memiliki 2 variasi yang mendapatkan namanya dari penemu rumus tersebut, yaitu Moore (Edward Forest Moore 1925-2003) dimana output hanya bergantung pada variabel sehingga tidak dapat mengulang state yang sama tanpa melakukan state yang lain (kunci pada output) dan Mealy (George H. Mealy 1927-2010) dimana output dapat dipengaruhi oleh variabel dan input sehingga sebuah state dapat dilakukan berulang apabila persyaratan pengulangan terpenuhi (kunci pada input). [20]

Kelemahan utama dari pola FSM adalah FSM tidak dapat melakukan aktivitas diluar pembuatannya (struktur pembuatannya infleksibel). Selain itu pula penanganan FSM juga susah diterjemahkan oleh mereka yang tidak berpengalaman dan tidak dapat diaplikasikan ke semua bidang pemrograman.[21]

Namun kelebihan utama dari FSM adalah desain utamanya yang lebih pesat dari *decision tree*, namun lebih sederhana dibandingkan automata yang kontemporeri (memiliki jenis yang serupa) karena FSM memiliki kode tahapan yang mudah dipahami namun tidak terlalu rumit dan penggunaan prosesor yang minimal.[21]

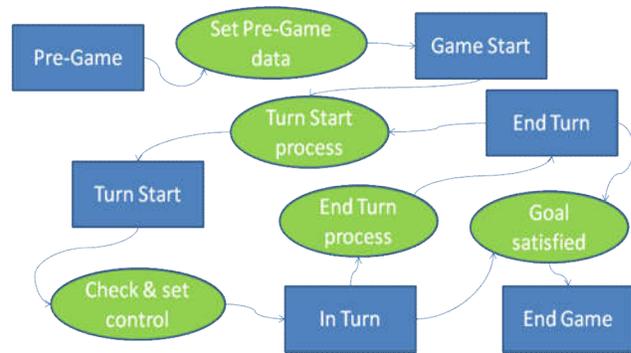
### 3. ANALISA DESAIN SISTEM

#### 3.1 Analisa Masalah

Salah satu masalah utama yang dihadapi dalam pemrograman kecerdasan buatan di dalam *video game* adalah seberapa mudahnya untuk AI dapat dieksploitasi apabila pola prosedurnya telah di"baca" dan diketahui oleh pemain. Ketika pola AI tersebut diketahui maka pemain akan dapat membaca dan memanfaatkan pola tersebut untuk mempermudah sesi permainan atau bahkan menghilangkan tantangan dalam permainan tersebut. Salah satu solusi dari permasalahan ini adalah memberikan AI cara untuk dapat beradaptasi dengan perilaku pemain seiring dengan berjalannya sesi permainan. Pilihan yang lain adalah memberikan variasi kepada AI untuk menjawab berbagai situasi yang ada dalam sesi tersebut.

Di dalam *Turn-Based Strategy game*, pada umumnya fokus utama dari AI adalah untuk membuat sebuah keputusan untuk menggerakkan bidak untuk melakukan sebuah tindakan yang terlihat untuk memberikan pemain sebuah tantangan sebelum pemain dapat menyelesaikan tujuan dari pihak pemain. Namun apabila pola AI dari game tersebut diketahui, maka pemain dapat melakukan sebuah tindakan diluar prosedur AI sehingga AI tidak dapat bereaksi tindakan pemain dan bertindak "tidak wajar". Hal ini akan menyebabkan permainan menjadi lebih mudah atau membosankan apabila pola AI terlalu sederhana untuk dibaca. Namun disisi yang lain, AI yang terlalu "curang" (dengan konteks antara AI mendapatkan keuntungan lebih daripada pemain atau memilih perhitungan kalkulasi probabilitas yang telah AI ketahui tingkat kesuksesannya dan selalu memilih aksi dengan keberhasilan tertinggi) dalam memilih aksi dapat mengakibatkan permainan menjadi terlihat tidak adil.

Dalam skripsi ini, akan diperlihatkan sebuah permainan *turn-based* melalui *unity* dengan AI yang memiliki pola yang berbeda berdasarkan data yang ada di dalam sesi permainan untuk dipilih oleh ID3 yang digunakan untuk memilih sebuah tindakan, namun AI diprogram untuk tidak dapat membaca probabilitas kesuksesan aksi sehingga "tampak" bahwa AI melakukan tindakan untuk beradaptasi dengan tindakan pemain tanpa keperluan untuk "mencurangi" permainan.



Gambar 3. Data flow permainan dari awal sampai akhir

#### 3.2 Jenis Cara Kerja Kecerdasan Buatan

Untuk dapat membuktikan pengaruh ID3 pada kecerdasan buatan dan pentingnya pengambilan data permainan, akan dibuat 3 cara kerja kecerdasan buatan pada permainan, yaitu cara kerja adaptif, cara kerja statik, dan cara kerja acak. Ketiga cara kerja ini akan menggunakan struktur kecerdasan buatan yang sama namun cara pengambilan keputusan yang berbeda kepada setiap cara kerja.

Setiap cara kerja kecerdasan buatan akan memiliki 3 pola yang akan dapat diambil sebelum AI melakukan tindakan pengambilan keputusan: Netral yang berjalan normal, Agresif yang berorientasi untuk menyerang lawan, dan Defensif yang berorientasi untuk mempertahankan diri. Namun pada AI yang bukan acak akan memiliki satu pola tambahan: Pola Ekonomik yang berfungsi untuk menghemat pengeluaran keuangan untuk disimpan kepada putaran selanjutnya.

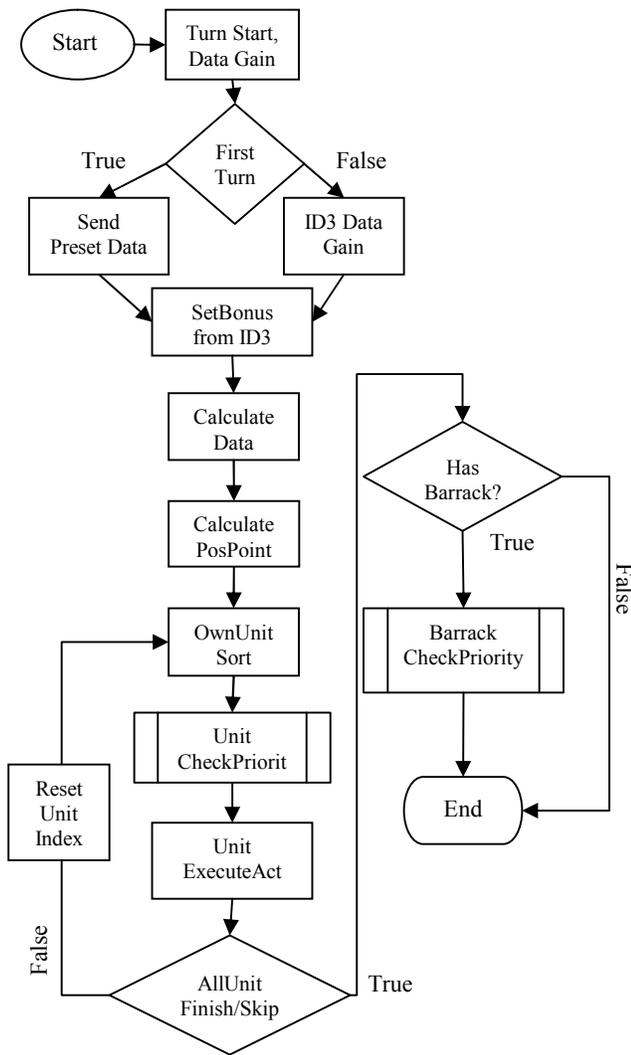
Pada putaran pertama dikarenakan data ID3 belum terbuat, maka AI diberikan data awal untuk dimasukkan ke data ID3. Dan setiap awal putaran akan dilakukan pemeriksaan untuk pengambilan pola.

Dan pengambilan pola ini diambil berdasarkan 2 faktor: Jenis AI yang dipilih dan data yang telah diambil pada awal putaran. Data meliputi statistik seperti seberapa banyak bidak yang dimiliki, situasi keuangan pada putaran tersebut, rasio antara keuangan yang dimiliki dan biaya bidak dan situasi bangunan yang ada.

Data yang telah diambil ini akan dianalisa untuk mengetahui situasi permainan. Lalu AI akan mengurutkan data data bidak dan bangunan yang dimiliki dan bertindak sesuai dengan data yang telah dianalisa ini. Dan AI akan menggerakkan bidak berdasarkan dengan pilihan posisi yang ada. Apabila ada bidak yang tidak memiliki posisi yang dipilih, bidak akan dilompati untuk dapat memilih keputusan lain setelah bidak bidak yang lain bergerak.

Setelah semua bidak bergerak atau semua bidak yang belum bergerak dilompati, maka AI akan memeriksa bangunan untuk membuat bidak sesuai dengan situasi keuangan yang dimiliki dan setelahnya AI akan mengakhiri putaran. Apabila uang yang dimiliki habis atau AI berada pada pola ekonomik, maka AI hanya akan membuat bidak yang murah sebelum mengakhiri putaran.

Setelah AI mengakhiri putaran, maka permainan akan berlanjut lagi sampai putaran AI yang sama muncul kembali dan prosesi ini diulang. Pada putaran kedua sampai akhir permainan, dikarenakan sudah ada data ID3, maka kalkulasi ID3 akan dilakukan. Kalkulasi ID3 akan mengambil data dan dihitung nilai entropy dan information gain apabila nilai data ada setidaknya dua pola yang berbeda.



**Gambar 4. Flowchart ringkas Kecerdasan Buatan adaptif dari awal ke akhir putaran**

Pada cara kerja adaptif, kecerdasan buatan pada awal putaran akan mengambil data pada permainan untuk menentukan pola apa yang sesuai dengan situasi permainan, dan pada awal putaran kedua akan menggunakan data tabel yang telah dibuat permainan untuk mencari nilai ID3 yang akan digunakan sebagai nilai bonus aksi sesuai dengan pola yang dipilih. AI adaptif menggunakan *random seed* untuk memilih maksimal tiga pilihan aksi dengan nilai prioritas tertinggi untuk mensimulasikan situasi bahwa kecerdasan buatan terlihat dinamik dan tidak statik. Namun apabila sebuah aksi memiliki nilai prioritas yang cukup mencolok dibandingkan aksi yang lain, aksi dengan nilai tertinggi akan selalu dipilih (dengan logika bahwa aksi tersebut akan memberikan keuntungan yang lebih jelas).

Cara kerja statik masih akan mengambil data awal putaran, namun data tersebut hanya digunakan untuk mendapatkan data bidak dan keuangan. Cara kerja ini akan selalu memilih aksi dengan nilai prioritas tertinggi dan tidak akan pernah memilih aksi yang berbeda atau menggunakan *random seed* untuk terlihat dinamik sehingga AI terlihat statik (selalu mengambil keputusan yang sama pada situasi yang sama).

Cara kerja acak akan selalu memilih aksi secara acak dengan memilih sebuah nilai secara acak untuk dipilih oleh program yang menggunakan cara kerja acak. Pada awal putaran, cara kerja ini akan memilih sebuah pola kecuali pola ekonomik secara acak. Dan menggunakan bidak sesuai dengan pola tersebut namun apabila sebuah bidak yang dimiliki memiliki lebih dari satu opsi, maka AI acak akan selalu memilih nilai secara acak menghiraukan nilai prioritas.

#### 4. IMPLEMENTASI

Pembuatan skripsi ini menggunakan Bahasa C# dengan Microsoft Visual C# 2010 sebagai Integrated Development Environment (IDE) bagi aplikasi ID3Editor dan sistem AI dan menggunakan bahasa C# dengan *MonoDevelop 4* sebagai IDE untuk permainan, dan *Unity* (atau *Windows Visual Application* berbahasa C#) sebagai media permainan. Alasan penggunaan C# untuk pembuatan aplikasi dan sistem AI adalah kemudahan dalam pembuatan Graphical User Interface (GUI) dikarenakan tersedianya komponen Windows Form yang dibutuhkan, tersedianya class-class bawaan di dalam framework .Net, dan kemampuan Unity untuk membaca Bahasa C#. Sedangkan alasan penggunaan MonoDevelop sebagai IDE untuk pembuatan game adalah MonoDevelop yang merupakan IDE bawaan Unity, sehingga proses debugging game dapat dilakukan dengan mudah melalui MonoDevelop. Unity digunakan sebagai media dalam pembuatan game dikarenakan kemudahan dalam membuat UI dan dalam mengimplementasikan lingkungan 3D.

Pemrograman aplikasi ini menggunakan prinsip pemrograman berbasis objek dan pemrograman prosedural. Aplikasi perangkat lunak memiliki 2 kelas, yaitu *AIClass* dan *ID3Class*, kedua kelas ini juga digunakan pada sistem kecerdasan buatan yang dipilih. Selain ID3, terdapat juga kelas *Game* dan *SubKelas* yang digunakan untuk sistem AI. Data-data yang disimpan oleh masing-masing Kelas dapat dilihat pada subbab berikut. Fungsi-fungsi pemilihan dari ID3 terdapat di dalam class *ID3Class*, sedangkan fungsi pemrosesan untuk kecerdasan buatan, sistem perhitungannya terdapat di dalam kelas *AIClass*, namun pengimplementasiannya di dalam game terdapat pada class *Game* (diluar class AI).

Pembuatan permainan komputer dengan metode Unity berbasis kepada objek dikarenakan Unity yang mengharuskan semua objek untuk dijadikan kelas tersendiri. Hal ini mengakibatkan permainan untuk memiliki kelas dan segmen fungsi yang berbeda beda. Setiap Objek di dalam permainan memiliki kemungkinan untuk memiliki kelas yang berbeda dimana data akan disimpan di dalam data di dalam Unity untuk dipanggil dan digunakan oleh skrip di dalam Unity itu sendiri.



**Gambar 5. Tampilan struktur permainan dan pengolahan data pada gambar berikutnya**

#### 5. UJI COBA SISTEM

Uji coba sistem akan dilakukan secara 2 tahap: Yaitu pengujian kecerdasan buatan dengan menggunakan dataset buatan untuk melihat hasil dari kecerdasan tersebut dan pengujian langsung

pada permainan yang meliputi sinergi antara kecerdasan buatan dengan ID3 dan kinerja variasi dari keputusan yang dibuat selama sebuah sesi permainan.

## 5.1 Ringkasan Tes Program

Isu dari program adalah program tidak dapat membaca data dengan nilai kosong karena akan menghasilkan error *NullPointerException*. Program dapat mengetahui apabila sebuah data merupakan data kembar karena data tersebut akan dihiraukan walaupun masih dibaca.

## 5.2 Ringkasan Hasil Uji Coba

Subbab ini akan menjelaskan secara ringkas bagaimana hasil uji coba sistem berdasarkan data pada putaran 1, 3 dan 5 dan ringkasan akhir permainan untuk ketiga jenis kecerdasan buatan berdasarkan percobaan repetisi sebanyak 3 kali

Perubahan yang terjadi pada putaran pertama hanya terlihat pada jenis bidak apa yang akan dibuat pada barak karena posisi tujuan untuk bidak yang dimiliki pada awal putaran akan relatif sama.

Pada putaran ketiga terdapat lebih banyak varian perbedaan daripada putaran pertama, namun tidak terlalu mencolok untuk data permainan memutuskan pola yang berbeda.

Semua percobaan pada kecerdasan buatan adaptif selalu memilih pola netral. Pola statik selalu memilih pilihan yang sama dan Pola acak menunjukkan output lebih bervariasi, namun akan terlihat jelas bahwa keputusan yang diambil AI tidak terlihat cerdas dalam pemutusan karena memang diambil hasil yang acak.

Pada putaran kelima, perbedaan yang terjadi pada putaran ketiga, keempat dan kelima akan memberikan variasi yang lebih berbeda daripada putaran ketiga dan konflik antar pihak sudah menjadi sebuah kepastian. Hal yang mempengaruhi hanyalah tingkat kerusakan yang terjadi antara konflik dan apa yang tersisa pada saat AI menjalankan putarannya.

Pada putaran kelima, pola AI adaptif akan memungkinkan untuk mengambil pola yang berbeda dan memilih aksi yang berbeda berdasarkan situasi bidak. Pola yang statik akan lebih berprioritas melakukan penyerangan dan tidak peduli apabila bidak sendiri lenyap akibat pertempuran. Dan pola yang acak... Terlihat cukup acak dalam memilih keputusan dan memiliki hasil output yang lebih berbeda namun tidak terlihat cukup cerdas karena terkadang AI acak akan memilih sebuah aksi yang kurang menguntungkan.

Dan apabila pada putaran kelima, AI adaptif memilih pola yang berbeda selain pola netral, maka AI akan memiliki nilai Information Gain ID3 yang berbeda pula. Dan nilai ini akan mempengaruhi beberapa input tergantung kepada pola yang dipilih.

Pada akhir permainan, ternyata AI adaptif yang menggunakan ID3 memberikan tantangan yang lebih dikarenakan waktu yang diperlukan untuk dikalahkan pemain memakan waktu paling lama. AI statik selalu menghasilkan permainan yang sama apabila dimainkan secara sama dan AI acak paling mudah dikalahkan karena pola ini tidak memikirkan logika dibalik alasan sebuah aksi sehingga paling mudah dikalahkan.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Berdasarkan pengaplikasian program, pengujian dan analisa data Dapat disimpulkan berbagai hal sebagai berikut:

1. Pengaruh dari ID3 pada program di awal putaran minim, dan bila AI hanya mengambil pola netral. Namun pengaruh ID3 akan mulai terlihat setelah awal permainan dimana nilai ID3 akan menjadi cukup signifikan untuk dapat memberi pengaruh kepada keputusan AI.
2. AI yang adaptif akan terlihat lebih dinamis dan cukup susah untuk dapat diterjemahkan dibandingkan dengan pola AI yang statik dan karena mengambil perhitungan berdasarkan kalkulasi dari data permainan, maka keputusan yang diambil juga lebih dapat dipahami daripada pola AI yang acak yang memilih aksi secara acak tanpa memperhitungkan keuntungan sebuah aksi.
3. ID3 dapat digunakan kepada tabel yang memiliki data dinamis (berubah seiring proses permainan berjalan). Namun tidak dapat berfungsi pada putaran pertama karena data tabel pada momen tersebut memang belum terbuat sehingga harus menggunakan nilai yang telah disiapkan secara manual (*Predetermined data*).
4. Kecerdasan buatan ternyata tidak dapat mempelajari data permainan secara dinamik sehingga apabila sudah difinalkan, maka kecerdasan buatan tidak akan dapat memilih aksi diluar kalkulasi pemrograman yang ada dan apabila terdapat kesalahan yang cukup fatal, maka keseluruhan program harus dirombak ulang.
5. Nilai titik bahaya dan titik kawan yang diterapkan untuk AI sangatlah berpengaruh karena hal tersebut dapat digunakan sebagai titik tujuan untuk AI dapat memberi perintah kepada sebuah bidak.

### 6.2 Saran

Beberapa dapat dijadikan saran dalam pengembangan selanjutnya:

1. Kemampuan program dalam menganalisa nilai null pada tabel data
2. Membuat proses pathfinding yang lebih ringkas tanpa perlu banyak komplikasi
3. Meningkatkan keefektifan ID3 dengan menggunakan pruning kepada program
4. Membuat AI yang dapat menyimpan kemampuan menganalisa data.
5. Peringkasan untuk beberapa cara AI mengambil keputusan

## 7. DAFTAR PUSTAKA

- [1] Adeyinka A., Adekanmi. O.O., Roseline. O., Samuel., and Sandika, Peter O. 2020 *Design and Implementation of an Intelligent Gaming Agent Using A\* Algorithm and Finite State Machines*. ORCID: 0000-0002-2592-2824 P.191 [http://www.ripublication.com/irph/ijert20/ijertv13n2\\_01.pdf](http://www.ripublication.com/irph/ijert20/ijertv13n2_01.pdf)
- [2] Baumgarte, R., Colton, S., Morris, M., 2008. *Combining AI Methods for Learning Bots in a Real-Time Strategy Game* <https://www.hindawi.com/journals/ijcgt/2009/129075/>
- [3] Crawford, C. 1991. *Computer Gaming World*. p. 78. Retrieved 18 November 2020 from <https://www.cgwmuseum.org/galleries/index.php?year=1991&pub=2&id=84>
- [4] Dannenhauer, D., Floyd, M.W., Decker, J., Aha, D.W. 2019. *Dungeon Crawl Stone Soup as an Evaluation Domain for Artificial Intelligence* <https://arxiv.org/abs/1902.01769>
- [5] E. Adams 2010. *Fundamentals of Game Design 2nd Edition*, Berkeley: New Riders.
- [6] El-Pro-Cus 2013. *Finite State Machine: Mealy State Machine and Moore State Machine*. Retrieved 3 April 2020

from <https://www.elprocus.com/finite-state-machine-mealy-state-machine-and-moore-state-machine/>

- [7] Ferruci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefel, N., and Welty, C. Building Watson 2010. *An overview of the DeepQA project. AI Magazine.*, 31(3):59–79.  
<https://ojs.aaai.org/index.php/aimagazine/article/view/2303>
- [8] Jazuli, H. 2018 *An Introduction to Decision Tree Learning: ID3 Algorithm*. Retrieved from 28 August 2020 from <https://medium.com/machine-learning-guy/an-introduction-to-decision-tree-learning-id3-algorithm-54c74eb2ad55>
- [9] Lancelot B., Matyas 2018. “*What is Finite State Machine?*” Retrieved 24 August 2020 from <https://medium.com/@mlbors/what-is-a-finite-state-machine-6d8dec727e2c>
- [10] Mr. CaveMan. 2012. *How to Integrate the Finite State Machine Compiler* Retrieved 28 December 2020 from <https://labs.grupow.com/blog/2012/03/27/how-to-integrate-the-finite-state-machine-compiler-with-cocos2d>
- [11] MotW1337. 2017 *Strategy: Turn-Based Strategy and Tactics* Retrieved 28 March 2021 from <https://motw1337.com/games/strategy/relationship-to-other-genres/turn-based-strategy-and-tactics/>
- [12] Mustofa, S., Rahmawati, E. 2018. *Penerapan Finite State Machine Untuk Pengendalian Animasi Pada Video Game RPG Nusantara Legacy*  
<http://jurnal.univbinainsan.ac.id/index.php/jusikom/article/view/251>
- [13] Peng, W.C.J., Zhou, H. 2015. *An Implementation of ID3 --- Decision Tree Learning Algorithm*  
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.5158>
- [14] Ramadhan, R., Widyani, Y. 2013. *Game Development Life Cycle Guidelines*. ICACSI 2013, pp. 95-100.  
<https://ieeexplore.ieee.org/abstract/document/6761558/>
- [15] Roedavan, R. 2014. *Unity: Tutorial Game Engine*; Penerbit Informatika Bandung
- [16] Rupali Kamble, and Prof. Deepali Shah. 2018. “*Applications of AI in Human Life*.” International Journal of Research - Granthaalayah, 6(6), 178-188.  
<https://doi.org/10.5281/zenodo.1302459>.
- [17] Sakkaf, Y. 2020 *Decision Trees: ID3 Algorithm Explained*. Retrieved from 25 August 2020  
<https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1?gi=4040b4cdda3a>
- [18] Tesauro, G. 1992. *Practical issues in temporal difference learning. Machine learning*, 8(3-4):257–277.
- [19] Tesauro, G. 1995. *Temporal difference learning and TD-Gammon. Communications of the ACM*, 38(3):58–68,
- [20] Tutorialspoint.com. Retrieved 24 August 2020 from [https://www.tutorialspoint.com/digital\\_circuits/digital\\_circuits\\_finite\\_state\\_machines.htm](https://www.tutorialspoint.com/digital_circuits/digital_circuits_finite_state_machines.htm)
- [21] Wilson, P., Mantooh, H.A. 2016. *Finite State Machines in VHDL and Verilog in Design Recipes for FPGAs (Second Edition)*; Zhang, Peng. 2008. *Application Software for Industrial Control in Industrial Control Technology*. Retrieved 24 August 2020 from <https://www.sciencedirect.com/topics/engineering/finite-state-machine>