

**IMPLEMENTASI *NON-PLAYER CHARACTER* PADA
GAME TURN-BASED STRATEGY MENGGUNAKAN
METODE *BEHAVIOR TREE***

SKRIPSI



**MOCH. DJIHAD AKBAR SYAH
H071181009**

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR**

2022

**IMPLEMENTASI *NON-PLAYER CHARACTER* PADA
GAME TURN-BASED STRATEGY MENGGUNAKAN
METODE *BEHAVIOR TREE***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains
pada Program Studi Sistem Informasi Departemen Matematika Fakultas
Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

MOCH. DJIHAD AKBAR SYAH

H071181009

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR**

2022

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Moch Djihad Akbar Syah

NIM : H071181009

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya dengan judul:

**Implementasi *Non-Player Character* Pada *Game Turn-Based Strategy*
Menggunakan Metode *Behavior Tree***

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa Sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 11. Oktober 2022

Yang menyatakan



Moch Djihad Akbar Syah

H071181009

**IMPLEMENTASI *NON-PLAYER CHARACTER* PADA *GAME*
TURN-BASED STRATEGY MENGGUNAKAN METODE
*BEHAVIOR TREE***

Disusun dan diajukan oleh:

**Moch Djihad Akbar Syah
H071181009**

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Menyetujui,

Pembimbing Utama

Dr. Hendra, S.Si., M.Kom.

NIP. 197601022002121001

Pembimbing Pertama

A. Muh Amil Siddik, S.Si., M.Si

NIP. 199110032019031

Ketua Program Studi

Dr. Hendra, S.Si., M.Kom.

NIP. 197601022002121001

Pada tanggal 11 Oktober 2022

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Moch Djihad Akbar Syah

NIM : H071181009

Program Studi : Sistem Informasi

Judul Skripsi : Implementasi Non-Player Character pada Game Turn-Based
Strategy Menggunakan Metode Behaviour Tree

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

Tanda Tangan

Ketua : Dr. Hendra, S.Si., M.Kom.

(.....)

Sekretaris : A. Muh. Amil Siddik, S.Si., M.Si.

(.....)

Anggota : Andi Muhammad Anwar, S.Si., M.Si

(.....)

Anggota : Edy Saputra Rusdi, S.Si., M.Si

(.....)

Ditetapkan di : Makassar

Tanggal : 11. Oktober 2022



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini dengan segala kerendahan hati penulis menyampaikan terimakasih yang tak terhingga kepada kedua orang tua penulis, Ibunda **Isoda Musa** dan Ayahanda **Drs. Syahrudin** sebagai tempat kembali setelah pergi, terima kasih atas kasih sayang, doa dan nasihat yang tulus sebagai bekal kehidupan. Penghargaan serta rasa terima kasih juga yang setulus-tulusnya kepada:

1. Rektor Universitas Hasanuddin Makassar **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc** dan seluruh Wakil Rektor dalam Lingkup Universitas Hasanuddin.
2. Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam **Dr. Eng. Amiruddin, M.Si** dan para Wakil Dekan serta seluruh staf yang telah memberikan bantuan selama penulis mengikuti Pendidikan di FMIPA Universitas Hasanuddin.
3. Bapak **Prof. Dr. Nurdin, S.Si., M.Si**, selaku Ketua Departemen Matematika FMIPA Unhas. Penulis juga berterima kasih atas dedikasi dosen-dosen pengajar, serta staf Departemen atas ilmu dan bantuan yang bermanfaat.
4. Bapak **Dr. Muhammad Hasbi, M.Sc** sebagai Ketua Program Studi Sistem Informasi Universitas Hasanuddin
5. Bapak **Dr. Hendra, S.Si., M.Kom.**, dan Bapak **A. Muh Amil Siddik S.Si., M.Si**, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
6. Bapak **Andi Muhammad Anwar, S.Si., M.Si** dan bapak **Edy Saputra Rusdi, S.Si., M.Si**, selaku dosen penguji terima kasih atas ilmu yang diberikan selama proses perkuliahan serta saran dan masukan yang diberikan dalam proses penyusunan skripsi ini.

7. Kanda **Rigel Rivaldo Subiyakto S.Kom** yang telah meluangkan waktu dalam membagi ilmu serta memberikan saran dan tambahan dalam pembuatan skripsi ini.
8. Teman-teman seperjuangan **Sistem Informasi 2018** tercinta yang selalu kebersamai, menemani, menguatkan dan menyemangati selama masa perkuliahan, serta terima kasih terkhusus saudara **La Ode Ahmad Khair A. S.Kom** yang turut membantu dan memudahkan penulis dalam penyusunan skripsi ini.
9. Saudara **Afwan Fahma Yusuf S.P** yang telah bersedia membantu penulis untuk menyediakan perangkat dalam rangka menyelesaikan proses penulisan skripsi ini.
10. **Nur Rahmasari** yang telah setia menemani, menguatkan dikala lemah, mengingatkan dikala lupa, serta meningkatkan motivasi untuk segera menyelesaikan skripsi ini. Semoga selalu dimudahkan segala urusan dan kesuksesan menunggu di depan mata, dan
11. Semua pihak yang telah memberikan bantuan kepada penulis baik berupa materi dan non-materi yang tidak dapat penulis sebutkan satu per satu, terima kasih untuk bantuan dan dukungannya.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Makassar, 11. Oktober 2022



Moch Djihad Akbar Syah

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK
KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan dibawah ini:

Nama : Moch Djihad Akbar Syah

NIM : H071181009

Program Studi : Sistem Informasi

Departemen : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

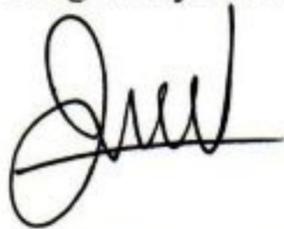
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

**Implementasi *Non-Player Character* Pada *Game Turn-Based Strategy*
Menggunakan Metode *Behavior Tree***

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilih Hak Cipta.

Yang Menyatakan



Moch Djihad Akbar Syah

ABSTRAK

Perkembangan ilmu pengetahuan dan teknologi kian pesat, salah satu jenis ilmu pengetahuan dan teknologi tersebut adalah perkembangan pada dunia *game*. Perkembangan pada industri *game* ini membuat perubahan dimana *game* juga dapat dijumpai dalam permainan berbasis komputer maupun *smartphone*. Berdasarkan perkembangan *game*, terjadi pula perubahan dan pengembangan dari segi jenis ataupun genre yang salah satunya adalah genre Turn-Based Strategy. Genre ini sangat digemari oleh sebagian besar masyarakat, bukan hanya karena mereka tumbuh bersama dengan jenis *game* tersebut namun manfaat dan hiburan yang diberikan pun sangat baik dan mudah untuk dipahami. Pada genre *game* ini pemain akan dihadapkan lalu pemain akan bergiliran satu sama lain untuk menyerang ataupun bertahan hingga terdapat pemenang dari salah satu pemain. Jika pemain tidak melawan pemain lainnya, maka yang akan dihadapi adalah NPC atau Non-Player Character yang dikendalikan sepenuhnya oleh komputer. Untuk menggerakkan sebuah NPC akan dibutuhkan sebuah algoritma atau metode, salah satunya adalah metode Behavior Tree. Sehingga akan dihasilkan sebuah *game* turn-based yang dapat dimainkan tanpa melawan pemain.

Kata Kunci: *Game, Behavior Tree, Turn Based, NPC*

ABSTRACT

The development of science and technology is growing these few years, one of them is *game* development. Developments in the *game* industry have made changes where games can also be found in computer-based games and smartphones. Based on the developments in terms of type or genre, one of which is the Turn-Based Strategy genre. This genre is very popular with most people, not only because they grew up with these types of games but because of the benefits and the entertainment they provide are also very good and easy to play. In this *game* genre, players will be confronted and then players will take turns with each other to attack or defend until there is a winner from one of the players. If the player doesn't fight with another player, then it will be faced by an NPC or Non-Player Character which is controlled completely by the computer. To move an NPC will need an algorithm or a method, one of which is the Behaviour Tree method. So it will produce a turn-based *game* that can be played without fighting players.

Keywords: *Game, Behaviour Tree, Turn Based, Non-Player Character*

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 <i>Game</i>	4
2.2 Turn-Based <i>Game</i> Strategy	6
2.3 Non-Player Character (NPC).....	7
2.4 <i>Behavior Tree</i>	7
2.5 Unity Engine.....	9
2.6 Metode <i>Waterfall</i>	11

BAB III METODE PENELITIAN.....	14
3.1 Waktu dan Lokasi Penelitian.....	14
3.2 Tahapan Penelitian	14
3.2.1 Analisis Kebutuhan.....	14
3.2.2 Perancangan Sistem.....	15
3.2.3 Implementasi Sistem.....	19
3.2.4 Pengujian	20
3.2.5 Hasil.....	20
3.3 Instrumen Penelitian	20
BAB IV HASIL DAN PEMBAHASAN	21
4.1 Analisis Kebutuhan Sistem.....	21
4.2 Perancangan Sistem.....	24
4.2.1 <i>Game Design Document</i>	24
4.3 Implementasi	34
4.3.1 Implementasi <i>Game</i>	34
4.3.2 Implementasi NPC (<i>Non-Player Character</i>).....	49
4.4 Uji Coba.....	52
4.4.1 Uji Coba <i>Gameplay</i>	52
4.4.2 Uji Coba NPC (<i>Non-player Character</i>)	57
BAB V KESIMPULAN DAN SARAN.....	60
5.1 Kesimpulan.....	60
5.2 Saran	60
DAFTAR PUSTAKA	62
LAMPIRAN.....	64

DAFTAR GAMBAR

Gambar 2.1 Final Fantasy Tactics: War of The Lions	6
Gambar 2.2 The Seven Deadly Sins: Grand Cross.....	7
Gambar 2.3 Contoh <i>Behavior Tree</i>	9
Gambar 2.4 Tampilan Awal Unity	10
Gambar 2.5 Metode <i>Waterfall</i>	12
Gambar 3.1 <i>Knight</i>	15
Gambar 3.2 Robot.....	15
Gambar 3.3 <i>Dino</i>	15
Gambar 3.4 Flowchart Alur <i>Game</i>	16
Gambar 3.5 <i>Start Scene</i>	17
Gambar 3.6 <i>Stage Choosing Scene</i>	17
Gambar 3.7 <i>Pick Character Scene</i>	18
Gambar 3.8 <i>Gameplay</i>	18
Gambar 3.9 <i>Player Lose</i>	19
Gambar 4.1 Alur <i>Gameplay</i>	21
Gambar 4.2 Alur <i>Behavior Tree</i> pada NPC.....	23
Gambar 4.3 <i>Chaos Realm</i>	25
Gambar 4.4 <i>Control Game</i>	26
Gambar 4.5 Informasi GUI.....	27
Gambar 4.6 <i>Knight</i>	27
Gambar 4.7 Robot.....	28
Gambar 4.8 RedHat	29
Gambar 4.9 Ninja	29
Gambar 4.10 Dino	30
Gambar 4.11 <i>Zombie</i>	30
Gambar 4.12 <i>Stage 1</i>	32
Gambar 4.13 <i>Stage 2</i>	32
Gambar 4.14 <i>Victory</i>	33
Gambar 4.15 <i>Defeat</i>	33
Gambar 4.16 <i>Main Menu</i>	34

Gambar 4.17 Animasi <i>Background</i> Sebelum dan Sesudah	35
Gambar 4.18 Animasi <i>Zombie</i> Berlari.....	35
Gambar 4.19 Animasi <i>Button</i>	36
Gambar 4.20 <i>Inspector</i> Animasi <i>Button</i>	37
Gambar 4.21 <i>Script</i> Animasi <i>Background</i>	37
Gambar 4.22 <i>Credit Scene</i>	38
Gambar 4.23 <i>Stage Choosing</i>	39
Gambar 4.24 <i>Introduction Stage</i>	39
Gambar 4.25 <i>Script</i> Berpindah <i>Platform</i>	40
Gambar 4.26 <i>Pick Character</i>	40
Gambar 4.27 <i>Template Scene Battle</i>	41
Gambar 4.28 <i>Template Card Slot</i>	42
Gambar 4.29 <i>Template Card Picked Slot</i>	42
Gambar 4.30 <i>Stage 1</i>	43
Gambar 4.31 <i>Stage 2</i>	43
Gambar 4.32 <i>Scenes</i>	44
Gambar 4.33 <i>Introduction Gameplay</i>	44
Gambar 4.34 <i>Canvas First Time</i>	45
Gambar 4.35 <i>Script First Time</i>	45
Gambar 4.36 <i>PlayerTurn</i>	46
Gambar 4.37 <i>BattleState</i>	47
Gambar 4.38 <i>EnemyTurn</i>	47
Gambar 4.39 Animasi Menyerang dari Karakter <i>Knight</i>	48
Gambar 4.40 Menyatukan <i>Sprite</i> untuk Menghasilkan Animasi Menyerang	48
Gambar 4.41 <i>Animator</i>	49
Gambar 4.42 <i>Floating Button</i>	49
Gambar 4.43 Karakter Lawan	50
Gambar 4.44 <i>Skill</i> Lawan	50
Gambar 4.45 Giliran Lawan	51
Gambar 4.46 List Kartu Pilihan Lawan Saat Ada Karakter dibawah 40% HP ...	51
Gambar 4.47 Terdapat Anggota <i>Party</i> Lawan Dibawah 40%	51
Gambar 4.48 Lawan Menggunakan Kartu <i>Heal</i>	52

DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Pada <i>Menu</i> Utama	52
Tabel 4.2 Hasil Pengujian Pada <i>Stage Choosing</i>	53
Tabel 4.3 Hasil Pengujian Pada <i>Character Pick</i>	53
Tabel 4.4 Hasil Pengujian Pada <i>Battle</i>	54
Tabel 4.5 Hasil Pengujian Pada Saat Defeat	56
Tabel 4.6 Hasil Pengujian Pada Saat Victory.....	57
Tabel 4.7 Hasil Pengujian Pada NPC	57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi kian pesat, salah satu jenis ilmu pengetahuan dan teknologi adalah perkembangan pada dunia *game*. Bermain *game* merupakan salah satu aktivitas yang sangat disukai oleh sebagian besar masyarakat. Dengan berkembangnya teknologi sekarang ini, *game-game* ini tidak hanya dapat kita jumpai pada kehidupan nyata, tapi dapat kita jumpai dalam permainan berbasis komputer maupun *smartphone*. Hingga saat ini, *game* kini bukan hanya digunakan sebagai alat hiburan namun *game* dapat digunakan sebagai mata pencaharian seperti *game development* dan bahkan sebagai profesi seperti atlet profesional, *content creator game*, dan sebagainya.

Menurut Putra dkk (2013) hingga saat ini *game* memiliki beragam jenis, mulai dari *game* yang hanya dapat dimainkan oleh satu orang saja hingga *game* yang dapat dimainkan oleh beberapa orang sekaligus. *Game* pun dapat dinikmati segala jenis usia baik anak-anak hingga orang dewasa. Hal ini disebabkan karena *game* bukan hanya sebagai media hiburan namun dapat digunakan sebagai alat untuk mengasah refleks dengan bermain *game rhythm*, mempelajari hal yang tidak dapat dilakukan di kehidupan nyata dengan *game* simulasi, serta mengasah kemampuan *problem solving* atau pengambilan keputusan dengan *game Turn-Based Strategy*.

Turn-Based Strategy Game merupakan *genre* pada sebuah *game* yang dimainkan secara bergiliran. Dalam praktiknya dua pemain akan saling berhadapan dan saling bergiliran untuk mengadu strategi satu sama lain untuk mengejar kemenangan. Beberapa *game* yang menggunakan konsep *Turn-Based Strategy* dalam *game* nya seperti Catur, Tic-Tac-Toe, Pokemon dan sebagainya. Dalam pembuatannya *game Turn-Based Strategy* biasanya hanya melibatkan satu pemain, yang kemudian akan menghadapi musuh dalam permainan berupa NPC (*Non-Player Character*). Oleh karena itu dibutuhkan kecerdasan buatan pada NPC sebagai lawan yang harus dihadapi *player*.

NPC merupakan karakter dalam *game* yang tidak dapat dimainkan oleh pemain dan berperan sebagai musuh atau lawan (Siswanto dan Suni, 2021). Asmiatun dan Hendrawan (2016) menjelaskan bahwa karakter *autonomous* atau yang disebut sebagai NPC adalah agen yang mewakili tokoh dalam sebuah cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka dimana *avatar* ini tindakannya berkebalikan dari *player* yang diarahkan secara *real time*. Segala aktivitas dari NPC dikontrol langsung oleh komputer dan pemain tidak dapat menginterupsi maupun mengubah perilaku yang telah ditentukan sebelumnya. Pemain hanya dapat mengikuti alur yang telah ditentukan pada sebuah NPC.

Sebuah *game* akan biasa saja jika NPC dalam *game* tersebut hanya menggunakan perintah *if* dan *else* dalam perilakunya, sehingga tidak ada tantangan serta pergerakan NPC tersebut akan lebih mudah untuk ditebak. Oleh karena itu dibutuhkan sebuah algoritma *decision making* yang berfungsi untuk mengatur perilaku dari NPC tersebut sehingga NPC tersebut akan memilih keputusannya sendiri berdasarkan kondisi yang dihadapi. Salah satu metode *decision making* yang dapat digunakan yaitu *Behavior Tree*. *Behavior tree* memiliki beberapa kelebihan yang tidak terdapat pada beberapa Algoritma *Artificial Intelligence* lainnya yakni algoritma ini dapat memberikan kemudahan pada proses pengembangan karena dapat dengan mudah membaca proses ataupun alur pengambilan keputusan (Junaidi dkk, 2021).

1.2 Rumusan Masalah

Rumusan masalah pada pembuatan *game Turn-Based Strategy* adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *Behavior Tree* ke dalam *game Turn-Based Strategy*?
2. Bagaimana merancang dan membangun *game Turn-Based Strategy* dengan metode *Waterfall*?

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mengimplementasikan metode *Behavior Tree* ke dalam *game Turn-Based Strategy*.

2. Merancang dan membangun *game Turn-Based Strategy*.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian yang dilakukan adalah menambah wawasan dan pengetahuan dalam pengimplementasian metode *Behavior Tree* dan bagaimana cara merancang dan membangun *Turn-Based Strategy game*.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. *Game* ini dikembangkan menggunakan *Unity* dan menggunakan bahasa C#.
2. *Game* ini dijalankan di komputer.
3. *Game* ini merupakan *game offline*.
4. *Game* ini hanya dimainkan oleh 1 orang (*Single Player*).
5. Penelitian hanya membahas desain *game* dan cara memprogramnya dengan menggunakan *Unity* tapi tidak membahas cara membuat *sprite* dan musik untuk *game*.

BAB II

TINJAUAN PUSTAKA

2.1 *Game*

Game merupakan sebuah kata dalam bahasa Inggris yang artinya adalah permainan. Permainan adalah suatu bentuk kegiatan yang terstruktur yang biasanya dilakukan untuk hiburan dan kesenangan dan kadang-kadang digunakan sebagai alat untuk mendidik. *Game* dapat juga diartikan sebagai aktivitas yang melibatkan satu atau lebih pemain yang memiliki tujuan yang ingin dicapai yang tentunya dibatasi oleh beberapa aturan. Pengertian *game* pada umumnya dapat berarti aktivitas yang bisa berupa tindakan nyata atau tindakan dalam suatu sistem yang dapat membawa kesenangan bagi pemain.

Menurut Tanjung (2013) *game* dapat diklasifikasikan menjadi beberapa jenis, yaitu berdasarkan *platform* yang digunakan, dimensi, dan *genre* dari *game* itu sendiri. Platform merupakan kombinasi spesifik dari komponen elektronik atau perangkat keras komputer dengan perangkat lunak yang memungkinkan *game* untuk beroperasi.

Berdasarkan platform yang digunakan, *game* dapat dibagi menjadi beberapa jenis, diantaranya:

1. *Console Game* yaitu *game* yang dimainkan menggunakan sebuah konsol tertentu seperti *Playstation 2*, *Playstation 3*, *XBOX 360*, dan *Nintendo Wii*.
2. *Arcade Game* yaitu *game* yang biasanya dimainkan pada sebuah mesin yang didesain khusus untuk beberapa *video game* tertentu yang biasa disebut ding-dong. Berbeda seperti konsol *game* yang hanya memiliki satu jenis *controller* saja, *arcade game* memiliki beberapa jenis *controller* seperti pistol, kursi khusus, dan setir mobil.
3. *PC games*, yaitu *game* yang dimainkan menggunakan komputer pribadi yang menggunakan kontroler berupa *mouse* dan *keyboard*.
4. *Handheld games*, yaitu *game* yang dapat dimainkan dimana-mana atau *portable* serta ukurannya yang lebih kecil dan ringan dibanding seperti *Arcade game* dan *Console Game* seperti *Nintendo DS* dan *Sony PSP*.
5. *Mobile games*, yaitu *game* yang dapat dimainkan khusus untuk *mobile phone*.

Tanjung (2013) mengatakan bahwa *genre* pada suatu *game* memperlihatkan pola umum tantangan dari *game* tersebut. Dengan perkembangan informasi seperti sekarang, *genre* dari *game* masih terus berkembang, sebagai contoh simulasi menari (*dance simulation*) yang diperkenalkan oleh desainer *game* dari Jepang. Namun secara umum *game* dapat dibagi menjadi beberapa jenis berdasarkan *genre* yang diterapkannya, yaitu:

1. *Role-Playing Games (RPG)*

Game RPG merupakan sebuah *game* mengenai konfigurasi dari karakter pemain yang meningkat berdasarkan *experience* dan jalan cerita (*storyline*) yang kuat. Dua elemen yang menjadi kunci suksesnya sebuah *game* RPG yaitu cerita (*story*) dan yang kedua adalah pembangunan karakter (*character development*).

2. *Game Olahraga (Sports Games)*

Game olahraga mempunyai tantangan yang tidak biasa bagi seorang desainer *game*. Tidak seperti *game-game* yang lain, dimana pemain memiliki sedikit pengetahuan mengenai dunianya, tetapi *game* olahraga meniru aturan pada olahraga di kehidupan nyata.

3. *Game Petualangan (Adventure Games)*

Game petualangan bukan merupakan sebuah kompetisi atau simulasi seperti *game* yang lainnya. *Game* jenis ini tidak menawarkan proses untuk dikelola atau mengalahkan musuh melalui strategi dan taktik. *Game* petualangan merupakan cerita interaktif mengenai karakter yang dikontrol oleh pemain.

4. *Game Puzzle (Puzzle Games)*

Game puzzle merupakan *game* yang bertujuan untuk memecahkan sebuah *puzzle*, terkadang tanpa menyatukan dengan jalan cerita atau tujuan yang lebih besar. *Game* jenis ini biasanya bervariasi pada satu tema saja.

5. *Game Aksi (Action Games)*

Game action merupakan *game* yang membutuhkan keterampilan seperti pengolahan informasi dan tindakan secara cepat. Hal ini memaksa pemain untuk membuat keputusan dan melakukan tanggapan dengan kecepatan yang jauh lebih besar.

6. Game Strategi (*Strategy Games*)

Pada *game* strategi biasanya pemain akan mengandalkan *decision making* dalam memainkan *game* strategi untuk menentukan hasil yang diinginkan. *Game* strategi dibagi menjadi 2 bentuk utama, yaitu *turn-based strategy* dan *real-time strategy*.

2.2 *Turn-Based Game Strategy*

Turn-Based Strategy Game merupakan turunan dari *game* strategi dimana pemainnya akan saling bergiliran pada pengambilan keputusannya dalam bermain. Pada *game* jenis ini, *player* dapat memainkan oleh satu pemain saja (*single player*) sedangkan lawannya adalah *computer* atau istilah lainnya disebut sebagai *Non-Player Character (NPC)*. *Game turn-based strategy* tidak memerlukan kecepatan dan keakuratan dalam bermain tetapi memerlukan pengambilan keputusan yang tepat yang dapat membawa pemain pada kemenangan. Varian *game Turn-Based Strategy* semakin berkembang, tidak hanya dalam bentuk catur namun ada pula yang berbentuk simulasi peperangan dengan aturan yang lebih kompleks, pemainnya masih bergiliran dalam mengambil keputusan (Utomo dan Widiastuti, 2013). Contoh *game Turn-Based Strategy* seperti pada gambar 2.1 dan 2.2.



Gambar 2.1 *Final Fantasy Tactics: War of The Lions*

(Sumber: CubeD3)



Gambar 2.2 *The Seven Deadly Sins: Grand Cross*
(Sumber: Netmarble)

2.3 *Non-Player Character (NPC)*

NPC atau *Non-Player Character* adalah objek karakter dalam permainan, yang dapat berupa manusia, hewan, robot, dll yang tidak dapat dikendalikan oleh pemain, tetapi dapat bertindak dan melakukan aktivitas yang tampaknya dikendalikan oleh pemain, tetapi sebenarnya NPC yang dikendalikan oleh program *computer*. Menurut Warpefelt dalam (Azzam, 2021) *Non-Playable-Character* atau *NPC* adalah objek dalam *game* yang bertindak sebagai karakter yang hanya dapat dikontrol secara terprogram dalam kondisi tertentu. *NPC* biasanya memiliki pola pergerakan atau kebiasaan seperti maju mundur, mengejar serta menyerang pemain.

2.4 *Behavior Tree*

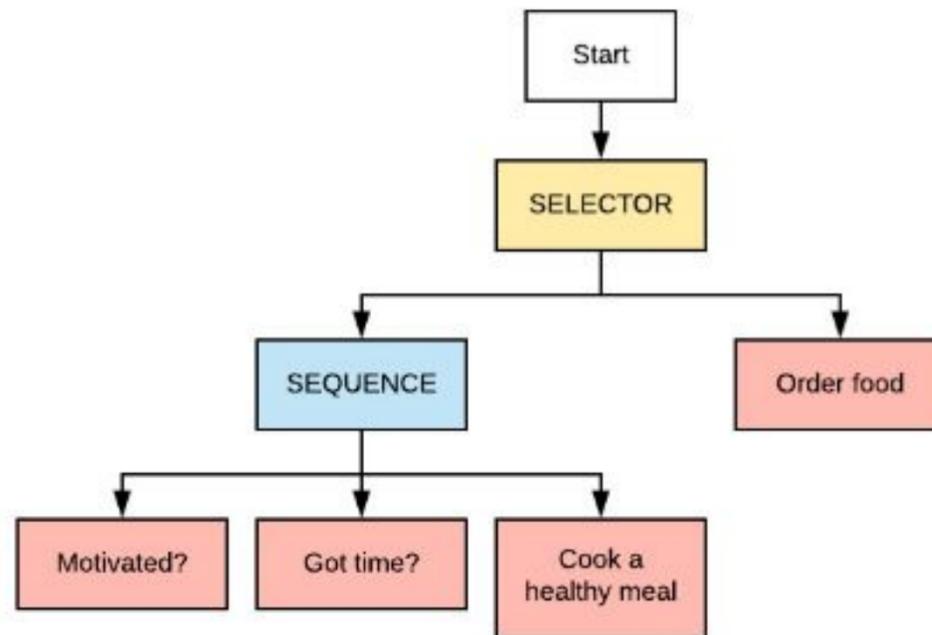
Behavior tree merupakan suatu model matematis berbentuk *tree* yang dibuat untuk merepresentasikan perilaku dari suatu entitas yang diprogram. *Behavior tree* adalah sesuatu yang sangat fundamental dalam pengembangan *game*. *Behavior tree* mirip seperti *finite state machine*, hanya saja tidak seperti

finite state machine yang berisi keadaan-keadaan, *behavior tree* berisi kumpulan instruksi dan perintah yang harus dijalankan oleh *developer* karena *behavior tree* mudah dimengerti oleh manusia sehingga tingkat kesalahan yang dibuat menjadi minimum (Christie, 2020).

Alur pemrosesan pohon pada *Behavior Tree* dimulai dari akar pohon (*root*), lalu tergantung jenis *node* pada akar, akan dilakukan salah satu atau semua aksi pada *node* (*node* proses) di cabang-cabang akar (Marcello, 2021). Lalu *node* yang diproses akan mengembalikan tiga status yaitu: *success* (berhasil), *failure* (gagal), *running* (diproses). Status-status ini menandakan hasil dari proses yang terjadi pada salah satu *node* pohon. Status berhasil memiliki arti proses yang dijalankan berhasil, gagal berarti proses yang dijalankan gagal, dan status *running* berarti proses yang sedang dikerjakan oleh *node* dan masih belum selesai dan ketika pohon dipanggil lagi oleh program, proses pada *node* akan dilanjutkan sampai mengembalikan status berhasil atau gagal.

Behavior tree terdiri dari beberapa jenis *node* yaitu *Sequence*, *Selector*, *Composite*, *Inverter*, *Repeater* dan *Leaf* (Soefana dkk, 2021). Setiap *node* dapat mengembalikan dari tiga status umum yang telah dijelaskan pada paragraf sebelumnya. Contohnya seperti pada gambar 2.3, *tree* dimulai dengan pada *root* kemudian setelah beralih ke *node* selanjutnya terdapat *node Selector* yaitu *node* ini akan menjalankan semua *child* yang dimulai dari *child* paling kiri hingga terdapat satu buah *child* yang mengembalikan nilai *success* sehingga *node selector* akan bernilai *success*. Kemudian pada *node selector* terdapat 2 *child* yaitu *node Sequence* dan *node Order food*, seperti yang telah dijelaskan mengenai *node Selector* yaitu setiap *child* akan dijalankan dari paling kiri sehingga *child* yang pertama dijalankan yaitu *node Sequence* yang akan mengembalikan nilai *success* jika seluruh *child* pada *node* tersebut bernilai *success*. Pada *node Sequence* terdapat 3 *child* yaitu *node* “motivated?”, *node* “Got Time?” dan *node* “Cook a healthy meal”. Ketiga *node* tersebut akan dijalankan sehingga salah satu dari mereka mengembalikan nilai gagal, jika ketiga *child* tersebut bernilai *success* maka *node Sequence* akan mengembalikan nilai *success* sehingga pada *node Selector* akan bernilai *success* karena sudah

terdapat *node* yang bernilai *success* yaitu pada *child node Selector* dan *node Order food* tidak akan dijalankan.



Gambar 2.3 Contoh *Behavior Tree*

(Sumber: Awaiting Bits)

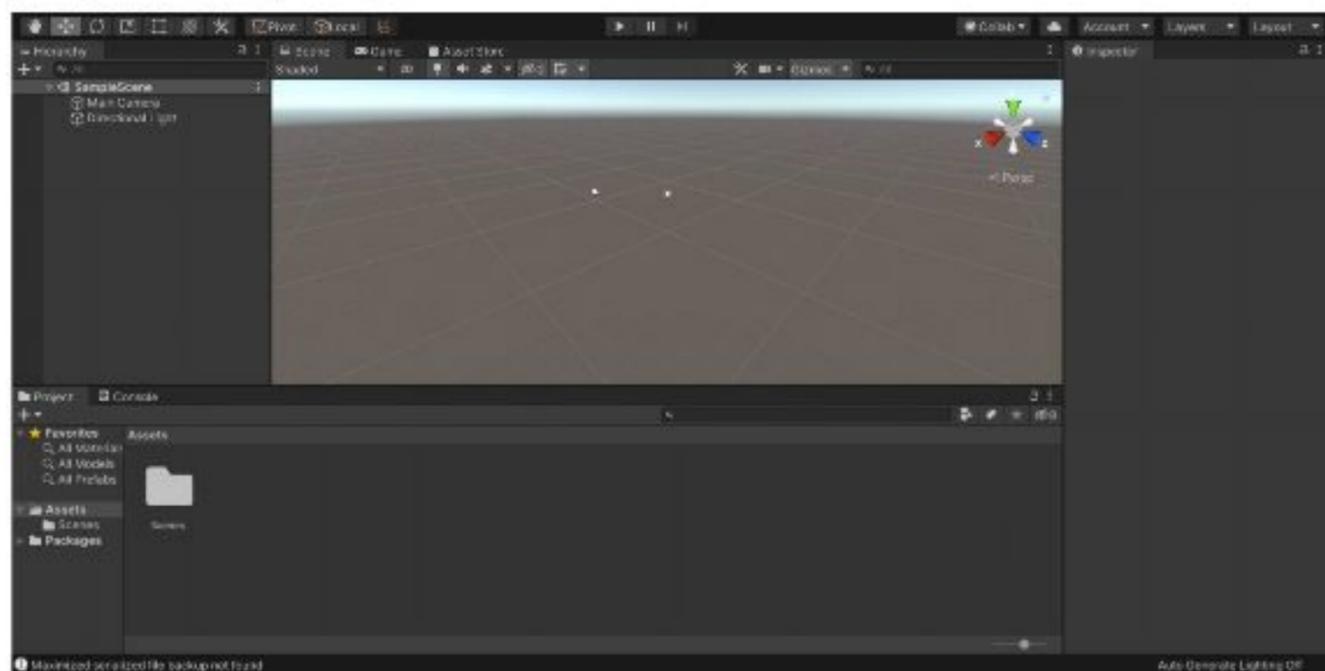
Adapun beberapa kelebihan dari *behavior tree* menurut (Rifqi, 2020):

1. *Behavior tree* merupakan metode yang elegan, sehingga memudahkan pengguna dalam mengatur *task*.
2. *Behavior tree* juga menggunakan alur desain umum sehingga penggunaannya tidak dibatasi hanya oleh *programmer* saja. Dengan alur desain tersebut juga dapat memudahkan dalam *debugging*.
3. Keunggulan dari *Behavior Tree* yang tidak dapat dipungkiri juga terdapat pada sifatnya yang modular, yaitu dapat digunakan berulang, dapat dihibridisasi dengan metode lain, dapat digunakan dalam banyak hal, tidak dibatasi hanya untuk *game*, serta memiliki kompatibilitas yang baik.

2.5 *Unity Engine*

Unity merupakan sebuah *game engine* yang berbasis *cross-platform* yang menurut Pratama (2018) adalah sebuah *software* yang dapat bekerja di hampir semua sistem dengan atau tanpa modifikasi. Mokodaser (2012) mengatakan bahwa *game engine* adalah sebuah sistem perangkat lunak yang dirancang

untuk pembuatan dan pengembangan suatu *video game*. *Game engine* memberikan kemudahan dalam menciptakan konsep sebuah *game* yang akan dibuat, Mulai dari *system rendering*, *physics*, suara, *scripting*, *Artificial Intelligent* dan bahkan *system networking*. *Game engine* dapat dikatakan sebagai jiwa dari seluruh aspek sebuah *game*. Tujuan digunakannya *game engine* adalah untuk mempermudah pembuatan bagian bagian tertentu dalam *game*, membagi-bagi pengembangan *game* menjadi modul-modul tertentu dan untuk memudahkan kolaborasi antar pihak (Mokodaser, 2012). Tampilan awal unity dapat dilihat pada gambar 2.4.



Gambar 2.4 Tampilan Awal Unity

Unity memiliki beberapa komponen utama yang digunakan dalam membuat *game* seperti *scene*, *C#*, *game object* dan *component*. Berikut penjelasan dari beberapa komponen *Unity*:

a. *Scene*

Scene merupakan *window* yang digunakan untuk membangun *game*. Di dalamnya kita bisa melihat dan mengatur *object* dalam sebuah *scene*. Dalam halaman resmi *unity*, *scene* adalah tempat kerja dan tempat mengatur objek-objek yang ada. *Scene* bekerja seperti *tab* pada *browser internet* dimana *game developer* dapat berkreasi dan meletakkan berbagai macam objek seperti rintangan, dekorasi didalamnya (Unity^a. 2022)

b. *C# (C-Sharp)*

C# (C-Sharp) merupakan sebuah Bahasa pemrograman yang dibuat oleh Microsoft dan salah satu Bahasa pemrograman yang dapat digunakan

dalam dunia pengembangan *game*. C# (*C-Sharp*) menjanjikan produktivitas, fleksibilitas serta kemudahan yang ada dari aplikasi sebelumnya yaitu *Visual Basic*, Java, dan C++ yang mengadopsi kemampuan dari penggabungan aplikasi sebelumnya (Yahya dan Nur, 2018:82).

c. *Game Object*

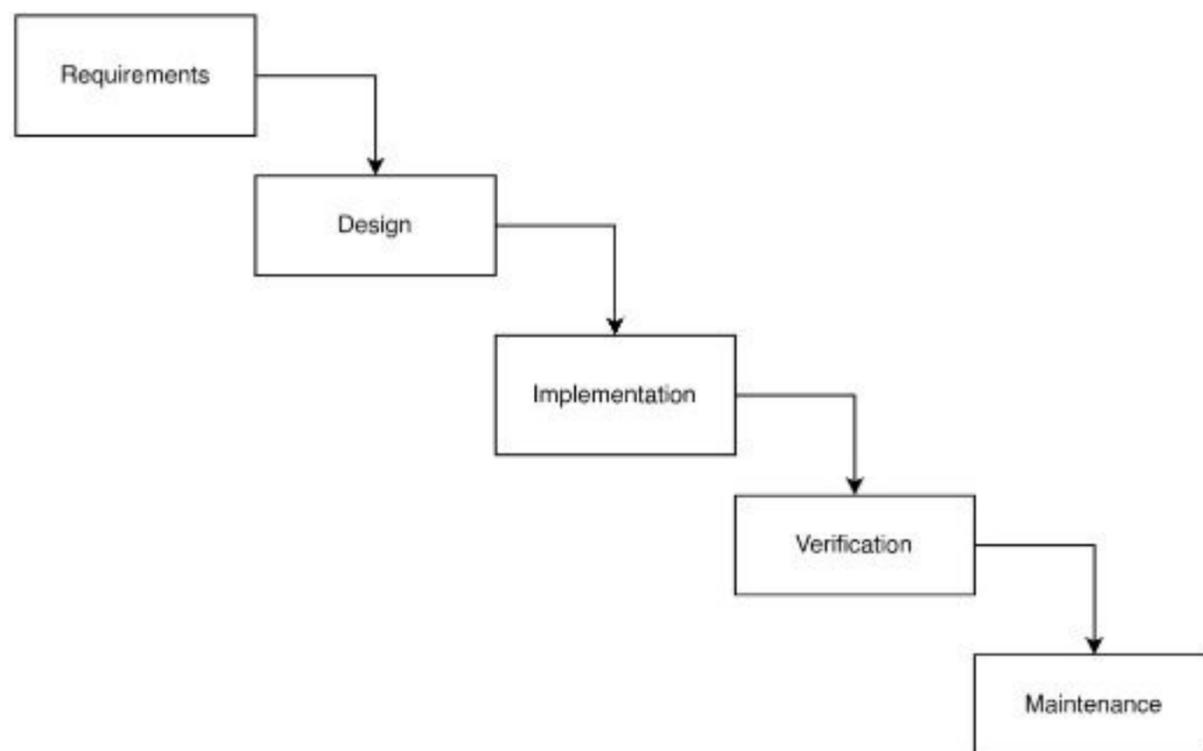
Dalam *unity* segala jenis objek yang ada disebut sebagai *game object*. Dalam Unity^b (2022) *Game object* merupakan sebuah *building block* pada *scene* di Unity, dan bertindak sebagai wadah untuk komponen fungsional yang menentukan bagaimana tampilan serta apa yang akan dilakukan oleh *game object* tersebut.

d. *Component*

Seperti yang dijelaskan sebelumnya pada *game object*, *game object* merupakan wadah dari *component*. *Component* yang dimaksud adalah bagian fungsional dari setiap *game object*. *Component* ini berisi properti yang dapat di edit untuk menentukan perilaku *game object* (Unity^c, 2022).

2.6 Metode *Waterfall*

Metode *waterfall* merupakan suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melawati fase-fase perencanaan, pemodelan, implementasi (kontruksi), dan pengujian. Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang runtut: analisis kebutuhan (*requirement*), desain sistem (*system design*), *coding* dan *testing*, penerapan program, pemeliharaan. Tahapan-tahapan tersebut dapat dilihat pada Gambar 2.5.



Gambar 2.5 Metode Waterfall

Berdasarkan tahapan-tahapan yang ada pada metode *waterfall*, berikut adalah penjelasan dari tahapan-tahapan dari metode *waterfall* menurut (Trisianto, 2022):

- a. *Requirement* (analisis kebutuhan)
 Dalam langkah ini merupakan analisis terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa dilakukan sebuah penelitian, wawancara atau studi literatur.
- b. *Design System* (desain sistem)
 Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada: struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural.
- c. *Coding dan Testing* (penulisan kode program/implementasi)
Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan nyata dalam mengerjakan suatu sistem. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang dibuat. Tujuannya adalah untuk menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

d. Penerapan program.

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

e. Pemeliharaan (*Operation and Maintenance*)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (*peripheral* atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.