

BAB II

TINJAUAN PUSTAKA

2.1 *Game*

Game merupakan salah satu bentuk hiburan yang berkembang pesat saat ini, dan bermain *game* telah menjadi aktivitas populer untuk mengisi waktu luang. Kemajuan dalam pengembangan *game* terus mengalami peningkatan, termasuk dalam pengembangan NPC, seperti penggunaan FSM untuk mengatur perilaku karakter musuh (Rohmah dkk., 2024). Salah satu genre *game* yang masih terus berkembang sampai saat ini adalah RPG. Genre ini masih banyak diminati oleh pemain karena elemen cerita yang begitu kental dilengkapi juga oleh watak dari masing-masing karakter yang mendalam (Mh dkk., 2020).

2.1.1 *Non-Player Character*

NPC dalam sebuah *game* merupakan salah satu komponen krusial yang mendukung terciptanya pengalaman bermain yang imersif. NPC memiliki peran dominan dalam menciptakan pengalaman imersif, terutama dalam konteks tantangan berbasis permainan (*challenge-based immersion*) (Carter dkk., 2021). NPC adalah karakter yang tidak dapat dikendalikan oleh pemain dan dapat berperan sebagai musuh, karakter netral, maupun karakter pendukung. Dalam beberapa penelitian, NPC tidak hanya berfungsi sebagai elemen naratif, tetapi juga sebagai entitas yang mampu berinteraksi secara dinamis dengan pemain melalui pendekatan kecerdasan buatan seperti *Finite State Machine* (FSM) (Andi dkk., 2023) dan algoritma pencarian jalur A* (Zulkarnain dkk., 2024). Dengan penerapan AI, NPC dapat menyesuaikan perilaku berdasarkan kondisi permainan dan keputusan

pemain, sehingga menciptakan dunia *game* yang lebih hidup. Bahkan, dalam konteks *metaverse*, pengembangan NPC dengan kemampuan percakapan berbasis AI semakin memperkuat peran NPC sebagai agen virtual yang adaptif dan sosial (Fegi, 2023).

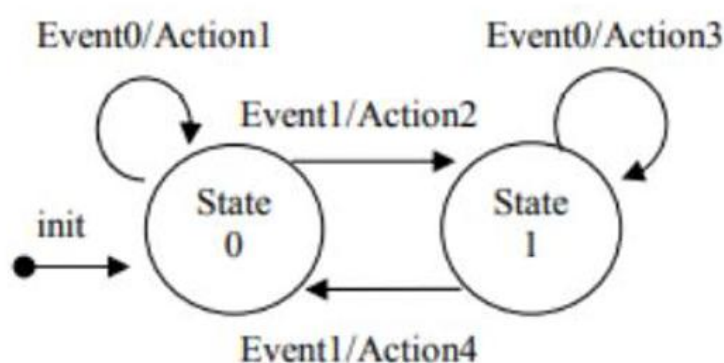
2.1.2 *Role-Playing Game*

RPG merupakan jenis permainan di mana para pemain mengambil peran sebagai karakter fiksi dan bekerja sama untuk membangun sebuah alur cerita. Dalam RPG, tidak ada konsep menang atau kalah yang mutlak, berbeda dengan permainan papan seperti Monopoli atau permainan kartu dan olahraga yang bersifat kompetitif. Layaknya sebuah novel atau film, RPG memiliki daya tarik tersendiri karena mendorong pemain untuk menggunakan imajinasi mereka dalam merancang cerita. Selain itu, RPG lebih menekankan aspek kolaborasi sosial dibanding persaingan, di mana para pemain biasanya tergabung dalam satu kelompok untuk mencapai tujuan bersama (Marzian dkk., 2020).

Penelitian oleh (Nguyen dkk., 2025) menyoroti bahwa RPG adalah bentuk penceritaan kolaboratif yang mendorong pemain untuk berinteraksi dan bekerja sama dalam menciptakan narasi, bukan sekadar mencapai kemenangan individu. Lebih lanjut, (Stubbs dkk., 2025) menemukan bahwa RPG dapat meningkatkan keterampilan sosial dan emosional, seperti pemecahan masalah kelompok, komunikasi interpersonal, dan regulasi diri. Selain itu, (Maisto dkk 2025) menunjukkan bahwa RPG berbasis kolaboratif memungkinkan pemain untuk menciptakan narasi bersama, dengan satu orang berperan sebagai *Game Master* yang memfasilitasi interaksi antar karakter pemain.

2.1.3 *Finite State Machine*

FSM digunakan untuk mengatur perilaku dari objek saat terjadi perubahan keadaan atau kondisi dalam permainan (Hormansyah dkk., 2020). Merupakan metode perancangan sistem kendali yang diterapkan untuk mengatur tindakan atau prinsip kerja suatu sistem dengan mengandalkan tiga elemen utama, yaitu *State*, *Event*, dan *Action*. Metode ini didasarkan pada model komputasi berupa mesin gabungan hipotetis yang terdiri dari satu atau lebih lokasi. Pada suatu waktu, hanya satu lokasi yang aktif, sehingga sistem harus beralih dari satu lokasi ke lokasi lainnya untuk menjalankan tindakan yang berbeda (Anastasya dkk., 2023).



Gambar 2. 1 *Diagram* FSM (Anastasya dkk., 2023)

Gambar 2.1 menunjukkan *diagram* FSM yang memiliki dua kondisi (*state*), dua masukan (*input*), dan empat tindakan keluaran (*output*) yang berbeda. Saat sistem dinyalakan, sistem akan beralih ke kondisi 0. Dalam kondisi ini, sistem akan menjalankan Tindakan 1 (*Action* 1) jika menerima masukan Kejadian 0 (*Event* 0), dan menjalankan Tindakan 2 (*Action* 2) jika menerima masukan Kejadian 1 (*Event* 1), lalu sistem akan beralih ke kondisi 1, dan seterusnya.

2.1.4 *Behavior Tree*

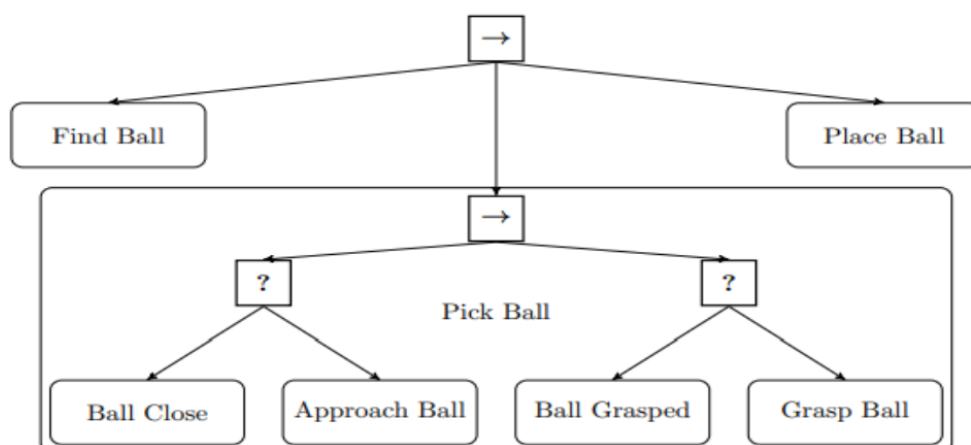
BT terdiri dari berbagai jenis *node*, di antaranya *node* tindakan yang menjalankan suatu aksi tertentu, *node* kondisi yang mengevaluasi apakah suatu persyaratan terpenuhi, serta *node* komposit yang menggabungkan beberapa *node* untuk membentuk perilaku yang lebih kompleks. Struktur modular ini mempermudah pengembangan perilaku yang rumit, memungkinkan pengembang *game* untuk dengan mudah menyesuaikan dan mengoptimalkan tindakan NPC. Hal ini meningkatkan interaktivitas pemain serta memperkaya pengalaman bermain secara keseluruhan (Zhu, 2025a).

Behavior tree memiliki berbagai macam elemen, tetapi elemen dasar yang harus ada dan pasti dimiliki oleh sebuah algoritma *Behavior tree* adalah sebagai berikut (Adrianus dkk., 2021b) :

1. *Leaf Node*, merupakan sebuah *node* yang bertugas menjalankan sebuah aksi. Seperti berlari, menyerang, berjalan, dan sebagainya.
2. *Condition component (if-else)*, merupakan sebuah komponen yang berupa nilai Boolean question, berfungsi mengembalikan nilai false atau true.
3. *Control flow (Selector & Sequence)*, adalah kumpulan komponen *child* yang berfungsi menentukan urutan pelaksanaan sebuah aksi, diantaranya terdapat dua macam control flow secara umum.

Dalam struktur BT, terdapat tiga jenis *node* utama: *root*, *control-flow*, dan *execution*. *Node* ini terhubung dalam *parent-child*. *Node root* adalah *node* teratas dan tidak memiliki *parent*. *node control-flow* memiliki satu *parent* dan setidaknya satu *child*, yang ditempatkan di bawahnya. *node execution*, di sisi lain, tidak

memiliki *child*, tetapi memainkan peran penting karena *node* inilah yang menjalankan tugas-tugas yang ditentukan dalam modul BT. Contohnya seperti pada gambar 2. 2.



Gambar 2. 2 Diagram Behaviour tree (Pratiwi et al., 2021)

Root node berperan sebagai pemicu dengan mengirimkan sinyal '*ticks*' ke *children node*, baik *node control-flow* maupun *node execution*, untuk memulai pelaksanaan tugas dalam BT. Setelah menerima sinyal '*ticks*', *node control-flow* akan menjalankan fungsinya dan meneruskan sinyal '*ticks*' ke *children node*. *Node control-flow* sendiri memiliki berbagai jenis dengan fungsi yang berbeda-beda.

2.1.5 Pengujian *Blackbox*

Pengujian blackbox dilakukan untuk mengevaluasi perilaku NPC musuh yang dikendalikan oleh algoritma FSM dan BT, dengan tujuan melihat apakah respon dan aksi NPC sesuai dengan yang diharapkan serta berjalan sesuai dengan rancangan algoritma (Izzatun dkk., 2024). fungsionalitas testing menjadi metode *black-box* yang relevan dalam penelitian yang dilakukan, metode ini secara spesifik berfokus pada verifikasi apakah setiap fungsi atau fitur yang dirancang untuk NPC,

seperti menyerang, bergerak, atau berinteraksi, bekerja dengan benar berdasarkan spesifikasi yang telah ditentukan (Azzam dkk., 2021). Fungsionalitas testing ini akan menitikberatkan pada interaksi fundamental antara *player* dan NPC mencakup validasi apakah NPC musuh mampu mendeteksi dan menyerang *player* secara tepat.

Penelitian yang mengembangkan BT secara evolusioner untuk *game* 3D dan menunjukkan bahwa BT dapat menghasilkan perilaku NPC yang sesuai dengan tujuan desain yang dilakukan oleh (Partlan dkk., 2022). Selain itu, studi komparatif antara FSM dan BT menunjukkan bahwa meskipun keduanya mampu mengendalikan perilaku NPC dengan efektif, modularitas dan pemeliharaan BT lebih unggul saat menangani kompleksitas yang lebih tinggi (Iovino dkk., 2024).

Penelitian lain juga mengonfirmasi bahwa implementasi FSM pada NPC dapat menghasilkan perilaku yang dinamis sesuai dengan kondisi permainan, seperti yang dijelaskan dalam studi simulasi kepemimpinan menggunakan FSM (Bagus dkk., 2024). Dengan demikian, pengujian *black-box* pada NPC berbasis FSM dan BT tidak hanya menilai fungsi dasar, tetapi juga mengukur adaptabilitas perilaku NPC.

2.1.6 Pengujian Akurasi

Untuk mengoptimalkan keunggulan masing-masing metode, algoritma FSM dan BT dapat dimanfaatkan sebagai arsitektur dalam pengendalian NPC musuh, di mana FSM menangani perilaku inti serta transisi state dasar secara cepat, sedangkan BT mengatur pengambilan keputusan yang lebih terstruktur sesuai konteks permainan. Pada sisi pengujian FSM dan BT, salah satu kriteria yang dapat

digunakan adalah Accuracy of State Transition (AST), yaitu pengujian yang menekankan agar seluruh transition pada FSM tercakup, sehingga setiap perpindahan state yang dirancang diuji minimal satu kali untuk memastikan kesesuaian perilaku dengan spesifikasi (Dwi Susanto & Arwin Dermawan, 2021).

2.1.7 Playtesting dengan Pendekatan Jakob Nielsen

Playtesting merupakan salah satu metode pengujian yang digunakan untuk menguji kualitas *game* yang telah dikembangkan. Dalam proses ini, pemain mencoba gim secara langsung lalu memberikan evaluasi berdasarkan pengalaman dan perasaan yang muncul selama bermain. playtesting memiliki beberapa variasi bentuk pengujian, salah satunya yaitu Fun Testin. Fun Testing berfokus pada pengukuran tingkat kesenangan (*fun factor*) dalam gim dengan mengamati respons atau reaksi pemain ketika memainkan *game* tersebut. Pemain yang terlibat dalam kegiatan playtesting disebut playtester. Secara umum, playtester dapat berasal dari lima kelompok, yaitu perancang gim itu sendiri, perancang gim lain, teman atau keluarga terdekat, penggemar gim, serta individu lain yang tidak memiliki hubungan langsung dengan pengembang (Fadillah dkk., 2021).

Pada penelitian (Hutauruk & Indra, 2024) menjelaskan bahwa pengujian dengan lima pengguna cukup untuk mendeteksi sebagian besar masalah, karena temuan baru menurun signifikan setelah 5 responden. Mereka juga menampilkan Model Poisson Nielsen/Nielsen-Landauer beserta rumusnya dan nilai tipikal $L \approx 31\%$. Dalam konteks penelitian ini, aspek yang diuji pada tahap beta playtesting bukan usability umum antarmuka, melainkan efektivitas penerapan algoritma FSM dan BT pada NPC. Oleh karena itu, playtesting terbatas pada pemain eksternal

digunakan untuk memvalidasi ketepatan transisi state, keputusan perilaku musuh, serta mendeteksi bug perilaku yang muncul dalam pengalaman bermain.

Perhitungan kelayakan *game* dilakukan dengan mengubah skor skala Likert menjadi indeks persentase. Indeks dihitung sebagai perbandingan antara skor aktual dan skor maksimum/ideal, kemudian dikalikan 100%, mengacu pada rumus yang digunakan pada penelitian-penelitian sebelumnya (Budi Hartanto dkk., 2025).

$$Indeks (\%) = \frac{\text{Total Skor Aktual}}{\text{Skor Maksimum}} \times 100\%$$

2.2 Penelitian Terkait

Penelitian yang berjudul “Perancangan *game* 3D *Being Healthy* Menggunakan UNITY 3D” menunjukkan hasil dari segi teknis, implementasi metode *Finite State Machine* (FSM) pada musuh di semua level berhasil diterapkan dengan baik, memungkinkan musuh untuk melakukan berbagai aksi sesuai dengan jarak pemain, seperti diam, mengejar, menyerang, dan mati ketika nyawanya habis. Hal ini menunjukkan bahwa *game* tersebut tidak hanya memiliki performa yang optimal tetapi juga memberikan pengalaman bermain yang interaktif dan memuaskan bagi pengguna (Nurien dkk., 2024).

Penelitian lain yang mengambil metode yang sama yang berjudul “Rancang dan Implementasi Metode *Finite State Machine* (FSM) Pada *Game Battle For Floryn* Berbasis Android” mendapatkan hasil yang menunjukkan bahwa penerapan FSM pada NPC lawan berhasil menciptakan perilaku yang cerdas. Dari 9 tombol yang digunakan dalam pengembangan *game*, semuanya berfungsi dengan sempurna. Penelitian ini melibatkan 10 responden yang menjawab 7 pertanyaan, dengan total faktor pembagi sebesar 70 (7 x 10). Hasilnya, 45 individu menyatakan

sangat setuju, 18 setuju, dan 11 tidak setuju. Secara persentase, 64,28% (45 orang) sangat setuju, 25,71% (18 orang) setuju, dan 15,71% (11 orang) tidak setuju dengan hasil yang diperoleh (Ellio dkk., 2024).

Pada penelitian yang dilakukan dengan menggunakan metode yang berbeda dengan judul “ Penerapan *Behavior Tree* Untuk Pengambilan Keputusan *Non-Player Character* Pada Gim Balap”. Hasil pengujian *Black Box* menunjukkan bahwa perilaku NPC berjalan sesuai dengan yang diharapkan. Dalam pengujian performa, gim tersebut mampu berjalan dengan rata-rata 60 FPS. Selain itu, semakin banyak NPC yang terlibat, semakin tinggi pula *frekuensi* tabrakan yang terjadi. Hal ini mengindikasikan bahwa gim dapat berjalan dengan FPS yang stabil, namun kemampuan NPC dalam mengambil keputusan menjadi lebih kompleks seiring dengan peningkatan jumlah NPC. Dengan demikian, dapat disimpulkan bahwa gim tersebut memiliki performa yang baik, tetapi tantangan pengambilan keputusan NPC meningkat ketika jumlah NPC bertambah (Soefana dkk., 2021).

Pada penelitian berikutnya, Implementasi *Behavior Tree* digunakan untuk menguji berbagai kondisi dan perilaku NPC musuh, meliputi beberapa *nodestate* utama seperti *Idle* (diam), *Patrol* (berpatroli), *Attack* (menyerang), dan *Death* (mati). Meskipun menggunakan *nodestate* yang sama, setiap NPC musuh memiliki karakteristik perilaku yang unik. Penggunaan metode *Behavior Tree* berhasil menciptakan perilaku NPC yang dinamis dan bervariasi sesuai dengan pemetaan yang telah dirancang sebelumnya (Hasti dkk., 2023).

Penelitian yang dilakukan terkait implementasi FSM pada *game* 3D *Adventure*, '*Dreadman*', untuk pengambilan keputusan karakter NPC, yang

menunjukkan keberhasilan penerapan FSM pada karakter musuh, meskipun dengan catatan bahwa 80% pengguna menilai tampilan *game* tersebut 'cukup' dan *game* berjalan lancar dengan minimal RAM 2GB. Penelitian lain menerapkan FSM pada *game side-scrolling 2D*, '*Adventure of Ruvy Fox*', yang menampilkan petualangan melawan monster, dengan hasil pengujian 100% berhasil untuk semua aspek permainan, termasuk pemain dan musuh, serta uji kegunaan SUS dengan skor 68,79 (kategori 'Ok'), yang mengindikasikan *game* dapat digunakan dengan baik oleh pemain (Rohmah dkk., 2024) dan (Suryo dkk., 2019.).

Pada penelitian yang mengimplementasikan BT dalam menentukan perilaku musuh pada *game Metroidvania "Cursed World"*. Penelitian ini menunjukkan bahwa BT dapat menghasilkan perilaku musuh yang bervariasi dan meningkatkan tantangan dalam *game*, yang diukur melalui *Game Experience Questionnaire* (GEQ) dengan rata-rata skor 4.05 (Alam & Haryanto, 2024). Penelitian selanjutnya juga meneliti penggunaan BT dalam perancangan perilaku bos musuh pada *game Action-Adventure* (Zelda-Like). Penelitian ini menghasilkan perilaku musuh yang interaktif dan responsif terhadap aksi pemain. Namun, hasil pengujian GEQ menunjukkan rerata 2.83, yang mengindikasikan bahwa pengalaman bermain yang diberikan masih perlu ditingkatkan. Namun, penelitian ini juga mengindikasikan perlunya pengembangan lebih lanjut untuk meningkatkan pengalaman bermain secara keseluruhan (Fauzan dkk., 2024).

Penelitian tentang mengembangkan algoritma kecerdasan buatan dengan mengimplementasikan *Collision Detection* dan FSM untuk mengatur perilaku karakter musuh pada *game* bertipe *Metroidvania*, serta menerapkan BT dan DSS

(*Decision Support System*) berbasis SAW (*Simple Additive Weighting*) untuk menentukan perilaku NPC secara dinamis. Kombinasi *Collision Detection* dan FSM memungkinkan karakter musuh merespon keberadaan pemain dengan lebih, sementara BT dan DSS berbasis SAW memberikan *fleksibilitas* dalam menyesuaikan perilaku NPC berdasarkan data interaksi pemain. Implementasi metode ini menghasilkan NPC yang lebih *adaptif* dan *responsif* terhadap perkembangan permainan, meningkatkan pengalaman bermain yang lebih interaktif (Farhan, 2023) dan (Alamsyah dkk., 2019) .

Pada penelitian yang berjudul “Rancang Bangun Prilaku Buatan pada *Non-Player Character* dalam *Game* Pemadam kebakaran Menggunakan *Finite State Machine* dan *Godot Script*” pendekatan *rule-based* diterapkan melalui metode FSM untuk mengatur perilaku NPC dalam *game* berbasis 2D. NPC dirancang memiliki beberapa kondisi seperti idle, patroli, dan menyerang, yang ditentukan oleh aturan transisi berdasarkan situasi dalam permainan. Penelitian ini menunjukkan bahwa pendekatan *rule-based* dengan FSM dapat menghasilkan perilaku NPC yang lebih terstruktur dan mudah dikontrol, serta mendukung terciptanya interaksi dinamis antara pemain dan karakter musuh. Evaluasi dilakukan secara kualitatif melalui pengamatan langsung terhadap respons NPC dalam berbagai skenario permainan, yang menunjukkan bahwa sistem berjalan sesuai dengan logika yang telah dirancang (Fikriansyah dkk., 2023b).

Pada penelitian dengan judul “*Boosting Cooperative NPC Effectiveness and Player Immersion through Behavior Tree Optimization in Gaming* “ dengan menggabungkan MCTS dan ANN, menghasilkan pemrosesan cepat sekaligus

pengambilan keputusan yang kompleks, sehingga mampu menciptakan NPC yang lebih responsif dan memperdalam keterlibatan pemain. Selain itu, penggunaan *Grammatical Evolution* (GE) untuk mengoptimalkan *Behaviour Tree* (BT) juga turut meningkatkan fleksibilitas dan efisiensi AI dalam *game*. GE memanfaatkan algoritma genetika untuk mengembangkan struktur BT yang lebih adaptif terhadap perubahan situasi dalam *game*. Dengan menyusun struktur BT menggunakan format XML, GE memungkinkan penciptaan pola perilaku yang kompleks secara otomatis. Ketika dikombinasikan dengan MCTS dan ANN, BT hasil evolusi GE memberikan pengembang alat yang kuat untuk menciptakan NPC yang cerdas dan mudah disesuaikan, sekaligus menyederhanakan proses pengembangan dan pemeliharaan AI dalam *game* (Zhu, 2025b).

Pada penelitian tentang implementasi *Behaviour Tree* pada NPC dalam *game Warik's Adventure* menunjukkan peningkatan signifikan terhadap pengalaman imersif pemain, yang dibuktikan melalui peningkatan skor pada *Game Experience Questionnaire* (GEQ). Berbeda dengan studi-studi sebelumnya yang belum secara eksplisit menghubungkan antara *Behaviour Tree* dan pengalaman imersif, penelitian ini memberikan kontribusi dalam pengembangan kerangka perilaku standar NPC berbasis *Behaviour Tree* yang mampu mendukung terciptanya pengalaman bermain yang lebih mendalam. Hal ini sangat penting, terutama dalam konteks *serious game*, di mana keterlibatan emosional dan kognitif pemain merupakan salah satu indikator keberhasilan. Ke depan, penelitian lanjutan dapat memperluas cakupan perilaku NPC yang diimplementasikan serta melibatkan

lebih banyak responden untuk meningkatkan validitas hasil evaluasi (Marchelputra dkk., 2023).

Pada penelitian mengenai implementasi *Finite State Machine* (FSM) pada karakter musuh dalam *game* menunjukkan bahwa metode ini berhasil diterapkan dengan baik. Hasil pengujian membuktikan bahwa karakter musuh mampu merespons interaksi pemain secara dinamis, mencerminkan perilaku yang sesuai dengan konteks permainan. Evaluasi melalui kuesioner menghasilkan skor usability sebesar 61,5%, yang dikategorikan dalam tingkat “baik”. Hal ini menunjukkan bahwa *game* yang dikembangkan dapat berfungsi dengan baik dan diterima secara positif oleh para responden (Ramadan 2022).

2.3 *State Of The Art*

State of the art membahas penelitian-penelitian yang relevan dengan penerapan FSM dan Behavior Tree dalam sistem kecerdasan buatan NPC pada game dan disajikan dalam Tabel 2.1.

Tabel 2. 1 *State Of The Art*

No	Peneliti & Tahun	Judul Penelitian	Metode / Algoritma	Jenis Game	Hasil Utama	Keterbatasan
1	(Nurien dkk, 2024)	Perancangan Game 3D <i>Being Healthy</i>	FSM	Game 3D (Unity)	FSM berhasil mengatur perilaku musuh berdasarkan jarak pemain (idle, chase, attack, death) dan meningkatkan interaksi pemain	Menggunakan satu algoritma (FSM)
2	(Ellio dkk, 2024)	FSM pada Game <i>Battle for Floryn</i>	FSM	Game Android	FSM menghasilkan perilaku NPC yang berjalan baik; mayoritas responden menyatakan sangat setuju	Fokus pada pengujian fungsional tombol
3	(Soefana dkk, 2021)	Penerapan Behavior Tree pada NPC Gim Balap	Behavior Tree	Game Balap	Perilaku NPC berjalan sesuai skenario dan stabil pada 60 FPS	Kompleksitas meningkat saat jumlah NPC bertambah

No	Peneliti & Tahun	Judul Penelitian	Metode / Algoritma	Jenis Game	Hasil Utama	Keterbatasan
4	(Hasti dkk, 2023)	Implementasi Behavior Tree pada NPC Musuh	Behavior Tree	Game Aksi	BT menghasilkan perilaku NPC dinamis meskipun memiliki node state yang sama	Tidak mengkaji kombinasi metode
5	(Rohmah dkk, 2024)	FSM pada Game 3D <i>Dreadman</i>	FSM	Game Adventure 3D	FSM berhasil diterapkan, namun tampilan game dinilai cukup oleh pengguna	Evaluasi berfokus pada visual
6	(Suryo dkk, 2019)	FSM pada Game <i>Adventure of Ruvy Fox</i>	FSM	Game 2D Side-Scrolling	Seluruh fungsi game berjalan 100%, SUS 68,79 (kategori OK)	FSM tunggal
7	(Alam & Haryanto, 2024)	Behavior Tree pada Game <i>Cursed World</i>	Behavior Tree	Metroidvania	BT meningkatkan variasi perilaku dan tantangan (GEQ 4,05)	Evaluasi pengalaman bermain masih terbatas
8	(Fauzan dkk, 2024)	Behavior Tree pada Game <i>Zelda-Like</i>	Behavior Tree	Action-Adventure	NPC interaktif dan adaptif terhadap aksi pemain	Skor GEQ relatif rendah (2,83)
9	(Farhan, 2023)	Implementasi Behavior Tree pada Game <i>Metroidvania</i>	Behavior Tree	Metroidvania	BT menghasilkan perilaku NPC yang dinamis dan adaptif terhadap kondisi permainan	Tidak menggunakan FSM

No	Peneliti & Tahun	Judul Penelitian	Metode / Algoritma	Jenis Game	Hasil Utama	Keterbatasan
10	(Alamsyah dkk, 2019)	FSM dan DSS pada NPC Game	FSM + DSS (SAW)	Game Petualangan	FSM mengatur perilaku dasar, DSS menentukan keputusan NPC	Tidak menggunakan BT
11	(Fikriansyah dkk, 2023b)	FSM pada Game Pemadam Kebakaran	FSM (Rule-Based)	Game 2D	FSM menghasilkan perilaku NPC terstruktur dan mudah dikontrol	Evaluasi kualitatif
12	(Zhu, 2025)	BT Optimization with MCTS & ANN	BT + MCTS + ANN + GE	Game Umum	BT menghasilkan NPC sangat adaptif dan kompleks	Implementasi sangat kompleks
13	(Marchelputra dkk, 2023)	Behavior Tree pada <i>Warik's Adventure</i>	Behavior Tree	Game Petualangan	BT meningkatkan imersi pemain (GEQ)	Cakupan perilaku terbatas
14	(Ramadan, 2022)	FSM pada NPC Musuh	FSM	Game Umum	FSM berhasil diterapkan, usability 61,5% (Baik)	Tidak membahas kombinasi metode

2.4 Matriks Penelitian

Matriks penelitian ini menjelaskan tentang perbedaan antara penelitian terdahulu dengan penelitian yang akan dilakukan. Dapat dilihat pada tabel 2.2.

Tabel 2.2 Matriks Penelitian

No	Peneliti	Algoritma			Platform		Genre		
		FSM	BT	Lainnya	Android	PC	RPG	FPS	RTS
1.	(Nurien dkk., 2024)	✓			✓		✓		
2.	(Ellio dkk., 2024)	✓			✓		✓		
3.	(Soefana dkk., 2021)		✓			✓			✓
4.	(Hasti dkk., 2023)		✓			✓	✓		
5.	(Suryo dkk., 2019)	✓				✓	✓		
6.	(Rohmah dkk., 2024)	✓				✓	✓		
7.	(Alam dkk 2024)		✓			✓	✓		
8.	(Fauzan dkk., 2024)		✓			✓	✓		
9.	(Farhan, 2023)		✓	✓		✓			✓

No	Peneliti	Algoritma			Platform		Genre		
		FSM	BT	Lainnya	Android	PC	RPG	FPS	RTS
10.	(Alamsyah dkk., 2019)	✓		✓		✓			✓
11.	(Fikriansyah dkk., 2023a)	✓				✓			✓
12.	(Zhu, 2025a)		✓			✓	✓		
13.	(Marchelputra dkk., 2023)		✓		✓		✓		
14.	(Ramadhanu dkk., 2022)	✓			✓			✓	
15.	Penelitian yang dilakukan	✓	✓			✓	✓		

2.5 Penelitian Terdekat

Berdasarkan tabel 2.2 Penelitian ini memiliki kedekatan dengan penelitian yang dilakukan oleh (Alam dkk, 2024) yang mengimplementasikan Behavior Tree dalam perancangan perilaku musuh pada game *Metroidvania Cursed World*. Hasil penelitian tersebut menunjukkan bahwa BT mampu menghasilkan perilaku musuh yang bervariasi dan meningkatkan tantangan permainan, yang dibuktikan melalui nilai rata-rata Game Experience Questionnaire (GEQ) sebesar 4,05. Meskipun demikian, penelitian tersebut juga menunjukkan bahwa struktur BT yang hierarkis cenderung menjadi lebih kompleks ketika jumlah kondisi dan perilaku musuh meningkat, sehingga berpotensi mengurangi efisiensi dalam pengelolaan transisi perilaku dasar.

Di sisi lain, penelitian oleh (Fikriansyah dkk., 2023b) yang menerapkan Finite State Machine (FSM) pada game Pemadam kebakaran, menunjukkan bahwa FSM efektif dalam mengatur transisi perilaku dasar NPC, khususnya pada karakter bos musuh yang membutuhkan respons cepat terhadap aksi pemain. Namun, hasil pengujian GEQ dengan nilai rata-rata 2,83 mengindikasikan bahwa perilaku NPC yang dihasilkan cenderung bersifat kaku dan kurang bervariasi, terutama ketika dihadapkan pada kondisi permainan yang dinamis dan kompleks.

Berdasarkan kedua penelitian tersebut, dapat disimpulkan bahwa FSM unggul dalam pengelolaan state dan transisi perilaku dasar secara sederhana dan efisien, tetapi memiliki keterbatasan dalam menangani variasi keputusan yang kompleks. Sebaliknya, BT mampu mengatur pengambilan keputusan secara hierarkis dan fleksibel, namun kurang optimal jika digunakan untuk mengelola

seluruh transisi state dasar NPC. Oleh karena itu, penggabungan FSM dan BT menjadi relevan, di mana FSM digunakan untuk mengelola state utama NPC, sedangkan BT berperan dalam mengatur logika keputusan di dalam setiap state. Pendekatan ini diharapkan dapat menghasilkan perilaku NPC yang lebih terstruktur, stabil, dan sesuai dengan kebutuhan pengendalian perilaku musuh pada game RPG yang menjadi fokus penelitian ini.