

BAB 2 LANDASAN TEORI

2.1 *Artificial intelligence*

Artificial intelligence secara sederhana adalah kemampuan mesin untuk meniru kecerdasan manusia dalam mengenali, memahami, dan mengambil keputusan, yang sangat penting agar robot dapat bekerja secara cerdas dalam berbagai situasi [9]. Kecerdasan mencakup kemampuan berpikir, membayangkan, mencipta, mengingat, memahami, mengenali pola, membuat keputusan, beradaptasi terhadap perubahan, serta belajar dari pengalaman [10]. Kecerdasan buatan berfokus pada pengembangan komputer agar dapat berperilaku seperti manusia dengan cara yang lebih mirip manusia [11], tetapi dalam waktu yang jauh lebih singkat dibandingkan manusia. Oleh karena itu, teknologi ini disebut sebagai Kecerdasan Buatan.

2.1.1 Algoritma A-Star

Algoritma A-Star merupakan teknik pencarian jalur yang efisien dalam menemukan rute terpendek antara dua titik pada grafik [12], algoritma ini sering diterapkan dalam permainan video dan robotika. Algoritma A digunakan untuk merencanakan jalur pada peta *grid*, dengan titik awal dan akhir sebagai input utama [13]. Algoritma ini beroperasi dengan mengembangkan simpul yang dianggap paling potensial dalam grafik berdasarkan estimasi heuristik terhadap jarak menuju tujuan [14]. Heuristik sendiri merupakan fungsi yang menghitung perkiraan jarak tersisa dari suatu *node* ke tujuan, yang kemudian digunakan oleh algoritma A-Star untuk menentukan urutan eksplorasi *node* berikutnya [2].

Salah satu keunggulan algoritma A-Star adalah kemampuannya dalam memastikan pencarian jalur terpendek apabila digunakan fungsi heuristik yang tepat [14]. Performa algoritma ini bergantung pada pemilihan fungsi heuristik, dan dengan heuristik yang efektif, A-Star dapat bekerja lebih cepat dibandingkan metode pencarian jalur lainnya [14]. Pendekatan ini berlandaskan pada persamaan 2.1 [2].

$$F(c) = G(c) + H(c); \quad (2.1)$$

H(c) menunjukkan jarak heuristik dari suatu *node* ke tujuan, sedangkan G(c) merepresentasikan panjang jalur dari titik awal melalui *node* yang dipilih. Heuristik adalah fungsi yang memperkirakan jarak tersisa dari suatu *node* ke tujuan. Algoritma A-Star memanfaatkan heuristik ini untuk menentukan prioritas eksplorasi *node* berikutnya [2]. Algoritma A-Star menentukan jalur optimal dengan memilih *node* yang memiliki nilai F(c) paling rendah. Keunggulan algoritma A-Star adalah fleksibilitasnya dalam menyesuaikan metode perhitungan jarak atau mengombinasikannya dengan faktor tambahan sesuai kebutuhan. Selain itu, algoritma A-Star memiliki keunggulan dalam kesederhanaannya untuk perencanaan jalur global statis, kecepatan dalam melakukan pencarian jalur, serta kemampuannya dalam memilih fungsi heuristik yang sesuai untuk masalah tertentu [15]. Hal ini menjadikannya sangat populer di bidang ilmu komputer dan penelitian robotika, seperti navigasi robot dan pengembangan *game* untuk menghasilkan pergerakan cerdas NPC [15].

Untuk nilai H(c) dapat diperoleh berdasarkan persamaan *Heuristic Manhattan Distance* (HMD) yang dapat dilihat pada persamaan 2.2.

$$d(a,b) = |x_b - x_a| + |y_b - y_a| \quad (2.2)$$

Pada Persamaan 2.2, d(a,b) merupakan jarak antara *node* a dan *node* b, di mana x menyatakan nilai absis dan y menyatakan nilai ordinat dari masing-masing *node*. *Manhattan Distance* menghitung jarak berdasarkan langkah horizontal dan vertikal [16]. Metode ini pas digunakan pada lingkungan *grid* yang pergerakan diagonal tidak diperbolehkan.

2.1.2 Perencanaan Jalur (*Pathfinding*)

Penelitian dalam bidang pathfinding atau pencarian jalur dalam game komputer telah mengalami perkembangan pesat, terutama karena kebutuhan akan pencarian jalur yang efisien dan *real-time* dalam *game* komersial. *Pathfinding* berfokus pada pencarian jalur paling pendek dari titik awal ke tujuan dengan menghindari rintangan di sepanjang jalur [17]. Berbagai algoritma telah dikembangkan untuk menyelesaikan permasalahan ini, seperti A-Star, Breadth-First Search (BFS), dan Depth-First Search (DFS), yang masing-masing memiliki pendekatan berbeda dalam menemukan jalur optimal.[18].

Dalam konteks perencanaan jalur berbasis komputer, berbagai algoritma telah dikembangkan dan umumnya dikelompokkan ke dalam dua kategori utama, yaitu algoritma pencarian klasik dan heuristik [17]. Algoritma pencarian klasik meliputi metode berbasis grafik seperti Depth-First Search (DFS), Breadth-First Search (BFS), dan Dijkstra, yang mengandalkan eksplorasi sistematis terhadap seluruh area pencarian [19]. Di sisi lain, algoritma heuristik menggunakan pendekatan estimasi untuk mempercepat proses pencarian jalur, dengan contoh populer seperti A-Star, D-Star, serta algoritma berbasis kecerdasan buatan seperti Algoritma Genetika (GA), Ant Colony Optimization (ACO), Jaringan Syaraf Tiruan (ANN), dan Simulated Annealing (SA) [19]. Pendekatan heuristik cenderung lebih efisien dalam skenario kompleks karena mampu memperhitungkan kemungkinan terbaik berdasarkan fungsi evaluasi atau heuristik tertentu. [20].

Perencanaan jalur dapat diklasifikasikan menjadi dua jenis berdasarkan sejauh mana informasi tentang lingkungan tersedia, yaitu perencanaan jalur global dan lokal [20]. Perencanaan jalur global menggunakan data lingkungan yang sepenuhnya diketahui untuk menentukan rute paling efisien, sehingga lebih sesuai diterapkan pada kondisi lingkungan yang statis. Sebaliknya, perencanaan jalur lokal digunakan dalam situasi di mana kondisi lingkungan bersifat dinamis atau tidak sepenuhnya diketahui, sehingga sering disebut sebagai perencanaan jalur dinamis. [20].

2.1.3 GUESS

Game User Experience Satisfaction Scale (GUESS) adalah instrumen psikometri berbasis skala Likert yang digunakan untuk mengukur kepuasan pemain terhadap *video game*. GUESS terdiri dari 9 subskala, yaitu: *Usability*, *Narratives*, *Play Engrossment*, *Enjoyment*, *Creative Freedom*, *Audio Aesthetics*, *Personal Gratification*, *Social Connectivity*, dan *Visual Aesthetics*, dengan total 55 pernyataan yang dinilai pada rentang 1 (Sangat Tidak Setuju) hingga 7 (Sangat Setuju). Penggunaan tiap subskala bersifat fleksibel, tergantung pada keberadaan elemen tersebut dalam game yang dievaluasi [21].

Berikut merupakan uraian dari masing-masing subskala dalam *Game User Experience Satisfaction Scale* (GUESS):

1. *Usability and Playability*

Menunjukkan sejauh mana *game* mudah dimainkan tanpa hambatan, baik

dari sisi kontrol maupun antarmuka pengguna, sehingga dapat mendukung pencapaian tujuan permainan secara optimal.

2. *Narratives*

Mengacu pada elemen cerita dalam *game*, termasuk alur peristiwa dan karakter yang dirancang untuk membangun ketertarikan serta memicu keterlibatan emosional pemain.

3. *Play Engrossment*

Mengukur kemampuan *game* dalam mempertahankan perhatian dan keterlibatan pemain secara konsisten selama sesi permainan berlangsung.

4. *Enjoyment*

Mencerminkan tingkat kesenangan dan kepuasan yang dirasakan pemain saat bermain *game*.

5. *Creative Freedom*

Menilai sejauh mana *game* memberikan kebebasan dan ruang bagi pemain untuk berkreasi serta mengekspresikan diri.

6. *Audio Aesthetics*

Meliputi kualitas aspek audio dalam *game*, seperti efek suara dan musik latar, yang mendukung suasana dan pengalaman bermain.

7. *Personal Gratification*

Menunjukkan sejauh mana *game* memberikan rasa pencapaian dan motivasi pribadi melalui berbagai tantangan atau penghargaan dalam permainan.

8. *Social Connectivity*

Menggambarkan kemampuan *game* dalam memfasilitasi interaksi dan konektivitas sosial antar pemain melalui fitur atau alat komunikasi yang tersedia.

9. *Visual Aesthetics*

Menilai daya tarik visual dan keindahan grafis yang disajikan oleh *game* dari sudut pandang pemain.

Namun, karena kuesioner awal terdiri dari 55 butir pertanyaan yang memerlukan waktu pengisian cukup lama, digunakan GUESS-18 sebagai versi ringkas dengan 18 butir pertanyaan yang mewakili sembilan subskala GUESS.

Instrumen ini memungkinkan pengisian dalam waktu 3–5 menit dengan tetap menjaga akurasi data, serta menggunakan rentang tujuh poin dengan definisi yang jelas pada tiap tingkatannya [22].

Setiap jawaban yang diberikan oleh responden akan memiliki skor tersendiri sesuai dengan nilai yang dipilih. Selanjutnya, skor-skor tersebut akan diolah untuk menghitung nilai rata-rata dengan menggunakan rumus tertentu.

$$Rata - rata\% = \frac{\sum_{i=1}^7 (n_i * i)}{n_{total} * 7} * 100\% \quad (2.3)$$

Angka n dalam rumus pada persamaan 2.3 di atas merujuk pada jumlah responden. n_i adalah jumlah responden yang memilih setiap pilihan jawaban i , yang berada pada skala 1 hingga 7 untuk pertanyaan GUESS yang diajukan. Nilai n_i kemudian akan dikalikan dengan skala jawaban masing-masing, di mana n_1 dikali 1, dan seterusnya hingga n_7 dikali 7. Hasil perkalian ini akan dijumlahkan, kemudian dibagi dengan total jumlah responden keseluruhan yang dikalikan dengan skala tertinggi, yaitu 7. kemudian dikalikan 100% untuk mengubah hasil menjadi persentase, sehingga mendapatkan rata-rata persentase yang menunjukkan seberapa tinggi nilai yang dipilih oleh responden dibandingkan dengan skala maksimum.

Tabel 2.1. Tabel Interval Penilaian GUESS

Rentang Persentase	Deskripsi
0% - 14%	Sangat Buruk
15% - 28%	Buruk
29% - 42%	Cukup Buruk
43% - 56%	Cukup
57% - 70%	Cukup Baik
71% - 84%	Baik
85% - 100%	Sangat Baik

Skala penilaian menggunakan persentase rata-rata dan hasil penilaian dapat diamati hasilnya berdasarkan dengan Tabel 2.1 di atas.