

BAB II. LANDASAN TEORI

2.1 *State of The Arts*

Beberapa penelitian yang dijadikan rujukan dalam penelitian ini adalah penelitian yang dilakukan oleh Novendra Setyawan dkk., pada tahun 2019 berjudul “Perencanaan Jalur Robot Sepak Bola Ummiros Menggunakan Algoritma A*” dari hasil pengujian menyatakan bahwa efisiensi jalur pada algoritma A* bergantung pada resolusi map yang dibuat. Jika dalam lingkungan statis maka resolusi rendah dapat digunakan dengan baik. Akan tetapi jika lingkungan dinamis maka efisiensi dari algoritma sangat diperlukan. Selain itu algoritma A* dapat menghindari halangan dengan baik dan menghasilkan jalur yang optimal (Setyawan et al., 2019).

Hasil penelitian yang dilakukan oleh Dedy Hermanto dkk., pada tahun 2018 yang berjudul “Penerapan Algoritma *A-Star* Sebagai Pencari Rute Terpendek pada *Robot Hexapod*” dari hasil pengujian Algoritma A* pada robot *hexapod* yaitu dapat membantu menemukan jalur terpendek untuk mencapai titik tujuan, tingkat kesuksesan robot untuk mencapai titik tujuan tanpa terkena halangan dengan rata – rata proses untuk mencapai ke titik tujuan yaitu paling besar 3 grid dengan tingkat kesalahan paling rendah 1 grid (Hermanto & Dermawan, 2018).

Hasil penelitian yang dilakukan oleh Bing Fu dkk., pada tahun 2018 yang berjudul “*An improved A* algorithm for the industrial robot path planning with high success rate and short length*” dari hasil pengujian menyatakan bahwa tingkat keberhasilan dari Algoritma *Improved A** sangat tinggi dibandingkan dengan Algoritma A*. Untuk rute yang diambil oleh Algoritma *Improved A** lebih pendek dibandingkan dengan Algoritma A*. Akurasi tingkat keberhasilan dari *Improved A** sendiri mencapai 80% dari sampel poin 100 dan 100% dari sampel poin 1200 (Fu et al., 2018).

2.2 **Dasar Teori**

Berdasarkan penelitian yang dilakukan maka terdapat beberapa landasan teori sebagai dasar penguat dalam melakukan penelitian sebagai berikut:

2.2.1 Robot Sepak Bola Beroda

Robot sepakbola merupakan salah satu jenis robot yang dikembangkan untuk keperluan perlombaan KRI (Kontes Robot Indonesia) kategori KRSBI (Kontes Robot Sepakbola Indonesia) Beroda. Robot Sepakbola Beroda dikembangkan agar dapat bermain seperti layaknya pemain sepak bola di lapangan, mulai dari mendeteksi bola, menggiring bola, menendang bola dan mencetak gol. Robot sepak bola menerapkan kecerdasan buatan (*Artificial Intelligence*) dan *Computer Vision* yang dapat mengenali objek berupa bola, perencanaan rute perjalanan (*Path Planning*), dan kinematika dan *control*. Robot sepak bola juga dilengkapi sensor kamera yang diletakkan dibagian depan untuk mendukung kemampuan bermain sepak bola (Julian Sahertian, 2018).

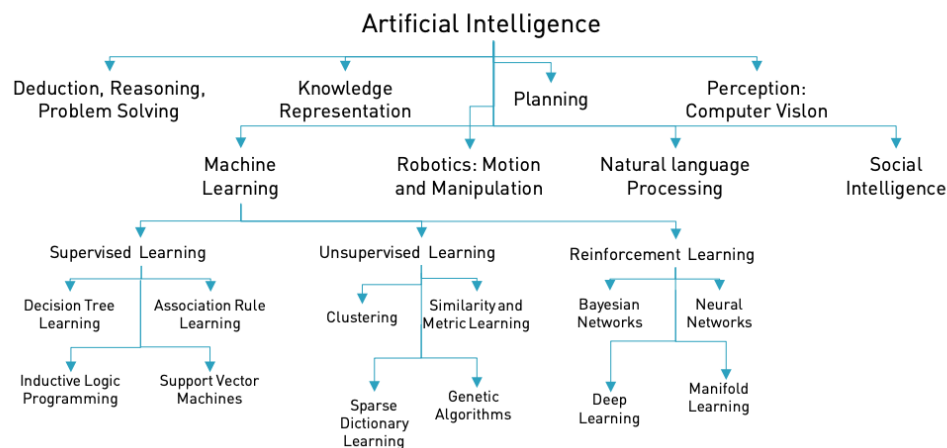


Gambar 2.1 Desain mekanik robot sepak bola beroda

Gambar 2.1 merepresentasikan objek yang digunakan untuk penerapan algoritma *Improved A-star search* (A^*) untuk perencanaan rute yang dilakukan.

2.2.2 *Artificial Intelligence*

Kecerdasan buatan atau *Artificial Intelligence* (AI) adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan. Gambar 2.2 merepresentasikan AI dari beberapa kelas. Masalah yang ditekankan pada penelitian ini ada pada kategori *Planning*. *Planning* didefinisikan sebagai pelaksanaan *output* dari hasil perencanaan yang melibatkan proses yang sesuai untuk representasi pengetahuan, pembuatan prioritas. Perencanaan dalam AI diterapkan dalam bidang robotika yang mengatur pergerakan sistem, penghindaran tabrakan, dan manipulator.



Gambar 2.2 Skema AI Sumber: (Providers, 2017)

2.2.3 Path Planning

Path Planning atau perencanaan jalur adalah tugas menemukan jalur berkelanjutan yang akan mendorong robot dari titik koordinat awal menuju titik koordinat tujuan. Dalam perencanaan jalur seluler, Sistem menggunakan peta lingkungan yang diketahui atau bersifat statis. Posisi koordinat robot akan direpresentasikan sebagai titik – titik yang mencakup semua kemungkinan robot terhadap rute perencanaan pada peta yang ditentukan. Robot bisa bergerak dari satu titik ke titik lain dengan menerapkan tindakan yang berbeda. Perencanaan rute akan menentukan jalur mana yang akan dilewati oleh robot berdasarkan algoritma yang diterapkan. Algoritma akan memilih status yang paling cocok untuk menentukan kemungkinan rute perjalanan yang dapat dikunjungi berdasarkan beberapa fungsi kriteria seperti jarak *heuristic* terpendek ke titik tujuan. Posisi koordinat saat ini menuju posisi koordinat titik tujuan mungkin memiliki satu atau lebih jalur yang dapat dilewati, atau bahkan tidak ada jalur sama sekali. Beberapa jalur yang dapat dilewati untuk mencapai titik tujuan, yaitu jalur yang tidak tertutupi atau terhalang oleh robot lawan atau *obstacle* (Klančar et al., 2017). Untuk mempersempit jangkauan yang dipilih dilakukan penentuan optimasi yang diinginkan antara lain:

1. Memilih pemilihan jalur yang paling pendek.
2. Memilih jalur yang cepat dilalui oleh robot.
3. Memilih jalur yang jauh dari halangan atau rintangan.
4. Memilih jalur yang mulus tanpa adanya belokan tajam.

5. Jalur harus mempertimbangkan kendala gerak (misalnya, kendala holonomik, dimana untuk semua rute perjalanan robot memungkinkan untuk dipilih).

2.2.4 Socket Programming

Socket merupakan salah satu teknologi dalam pemrograman jaringan komputer. Dengan *socket*, aplikasi jaringan dapat saling berkomunikasi sesuai mekanisme standar yang terdefinisi pada sistem operasi dan perangkat jaringan (Kusumo et al., 2020). Sebuah *socket* merepresentasikan koneksi tunggal antara dua *end host*, sehingga disebut pula dengan koneksi *point-to-point* (Kusumo et al., 2020). Lebih dari satu *socket* juga dapat dibuat dalam lebih dari satu aplikasi. Selain menghubungkan dua atau lebih *host* yang berbeda, *socket* juga dapat digunakan pada komunikasi lokal dalam satu komputer yang sama. Sifat dari *socket* sendiri adalah *bidirectional* atau komunikasi dua arah, sehingga suatu *host* dapat mengirim sekaligus menerima data (Kurose et al., 2017).

2.2.5 UDP (User Datagram Protocol)

UDP (*User Datagram Protocol*) adalah protokol komunikasi alternatif untuk TCP (*Transmission Control Protocol*). Tujuan dari protokol UDP adalah untuk menciptakan komunikasi dengan paket berukuran *header* kecil dan cepat (Kurose et al., 2017). Keunggulan dari UDP yang cepat dikarenakan UDP menerapkan komunikasi *connectionless oriented* (Kurose et al., 2017). Berdasarkan pada orientasi tersebut, antar *host* yang terhubung tidak perlu menjalin koneksi sebelum mengirimkan paket data sesungguhnya. Dengan demikian, paket data UDP atau yang juga dikenal dengan datagram dapat ditransmisikan secara langsung. DCCP adalah protokol yang berbasis UDP. Protokol DCCP ini menyediakan kontrol terhadap kemacetan tanpa menjamin kehandalan pengiriman data (Kurose et al., 2017). Dengan demikian, protokol ini menghubungkan kesenjangan antara TCP yang menyediakan pengiriman yang handal dengan kontrol kemacetan dan UDP yang tidak memberikan pengiriman yang handal. DCCP biasanya digunakan untuk aplikasi di mana data lama secara substansial kurang berharga daripada data terbaru (Kurose et al., 2017). Contohnya adalah VoIP (*Voice over Internet Protocol*) di mana jika suatu paket hilang, aplikasi tidak perlu transmisi ulang.

2.2.6 *A-star Search (A*)*

A-star search (A)* adalah pencarian umum yang dapat digunakan untuk menemukan solusi untuk banyak permasalahan, salah satunya adalah perencanaan rute perjalanan. Algoritma *A** adalah algoritma paling populer dan secara luas menggunakan AI *pathfinding* untuk penyelesaian permasalahan pencarian rute yang diusulkan oleh Hart, Nilsson dan Raphael pada tahun 1967. Dengan kesederhanaan dari algoritma ini, *A** hampir selalu menjadi pilihan terbaik untuk pencarian rute, karena menemukan jalur terpendek dari rute perjalanan yang akan dilewati. *A** menggunakan fungsi *heuristic* untuk meningkatkan perilaku yang terkait dengan Algoritma *Dijkstra* (Mathew, 2015). Algoritma *A** menemukan rute jalur menuju titik tujuan target dengan meminimalkan fungsi biaya minimum yang ditunjukkan pada fungsi yang ditunjukkan pada persamaan 2.1.

$$f(cost) = h(n) + g(n) \quad (2.1)$$

dengan $h(n)$ adalah pencarian jarak *heuristic* terhadap posisi saat ini terhadap posisi tujuan akhir, sedangkan $g(n)$ merupakan jarak *heuristic* pencarian posisi saat ini terhadap posisi tujuan selanjutnya. Jarak *heuristic* direpresentasikan pada persamaan 2.2 dengan fungsi biaya *euclidean distance* dan persamaan 2.3 dengan fungsi biaya *manhattan distance*. Pemilihan fungsi biaya *heuristic* yang digunakan merupakan metode yang populer dalam perencanaan rute menggunakan metode *A-star Search (A*)* (Prasetyo et al., n.d.).

2.2.7 *Euclidean Distance*

Euclidean Distance adalah metode pengukuran jarak terdekat antara dua titik dalam ruang *Euclidean*. Metode ini biasanya diterapkan pada perhitungan dua dimensi dan tiga dimensi, tetapi dapat dikembangkan ke dimensi yang lebih tinggi (Nurliza, 2018). Jika diandaikan terdapat dua titik (a, b) dalam bidang dua dimensi, dengan koordinat a adalah (ax, ay) dan koordinat b adalah (bx, by) , maka jarak antara titik a dan b dalam ruang *Euclidean* adalah akar kuadrat dari jumlah kuadrat dari perbedaan antara koordinat titik a dan b . Rumus dari *euclidean distance* direpresentasikan pada persamaan 2.2.

$$D(a, b) = \sqrt{((bx - ax)^2 + (by - ay)^2)} \quad (2.2)$$

Persamaan 2.2 secara teori sama seperti rumus segitiga *Phytgoras*, hal ini terjadi karena persamaan *Euclidean* mencari jarak terdekat antara dua titik dengan membuat garis bayang antara koordinat x dan koordinat y .

2.2.8 Manhattan Distance

Manhattan Distance merupakan sebuah dasar dari penentuan sebuah posisi atau lokasi relatif dari sebuah informasi jarak pada sebuah titik tertentu. Metode ini disebut blok absolut atau lebih dikenal dengan *city block distance* karena mengganti perbedaan kuadrat dengan menjumlahkan perbedaan absolut dari variabel (Nugraheny, 2017). Jika diketahui $u = (x_1, y_1)$ dan $v = (x_2, y_2)$ adalah dua titik koordinat, maka jarak kedua titik tersebut menurut metode *Manhattan Distance* dapat dihitung dengan persamaan 2.3.

$$D(u, v) = |x_1 - x_2| + |y_1 - y_2| \quad (2.3)$$

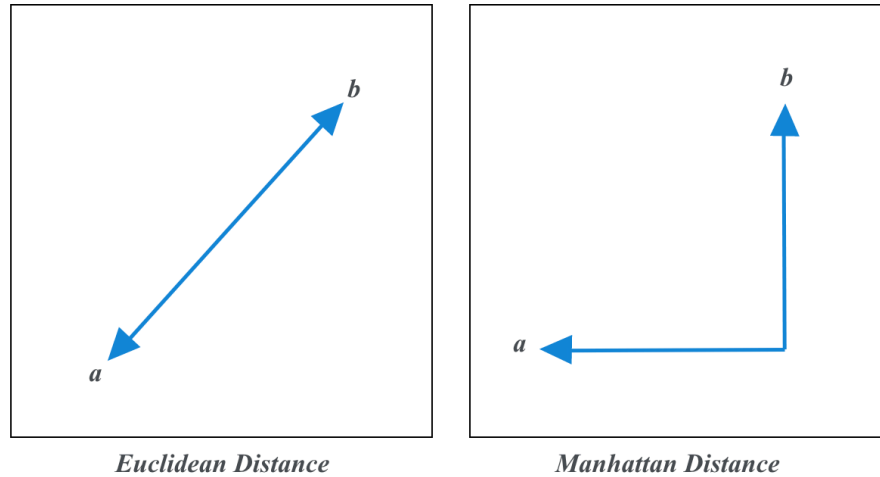
Namun ketika sebuah titik memiliki n dimensi, seperti $u = (x_1, x_2, x_3, \dots, x_n)$ dan $v = (y_1, y_2, y_3, \dots, y_n)$ maka persamaan 2.3 bisa diubah sesuai dengan jarak antara titik u dan v yang direpresentasikan pada persamaan 2.4.

$$\begin{aligned} D(u, v) &= |x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n| \\ &= \sum_{i=0}^n |x_i - y_i| \end{aligned} \quad (2.4)$$

2.2.9 Perbandingan Metode *Euclidean Distance* dan *Manhattan Distance*

Perbedaan metode *Euclidean* dan *Manhattan* terletak pada persamaan matematis antara keduanya. Secara sederhana jarak *Euclidean* adalah jarak terdekat dari titik a dan b dengan menggunakan persamaan segitiga *Phytgoras*, sementara jarak *manhattan* adalah penjumlahan absolut atau mutlak antara koordinat x dan koordinat y . Metode *Manhattan Distance* menghasilkan output jarak yang lebih panjang daripada metode *Euclidean Distance* karena metode *Manhattan* tidak

mengukur langsung jarak a ke b tetapi melewati garis koordinat x dan y . Gambar 2.3 merepresentasikan ilustrasi mengenai perbedaan antara jarak *Manhattan* dan *Euclidean*.



Gambar 2.3 Ilustrasi Perbedaan *Euclidean* dan *Manhattan*

2.2.10 *Improved A-star Search (A*)*

Perencanaan rute *Improved A-star search (A*)* termasuk perencanaan probabilistik. Perencanaan probabilistik terdiri dari dua fase yaitu, fase *pre-processing* dan fase *query*. Pada fase *pre-processing* rute perjalanan tanpa tabrakan dihasilkan secara acak dalam ruang konfigurasi robotik, dan dideteksi oleh algoritma *Hybrid Bounding Volume Hierarchy Tree*. Selama fase *query* dari perencanaan peta jalan probabilistik, jalur tanpa tabrakan dibangun antara titik – titik pengambilan sampel oleh perencanaan jalur lokal. Perencanaan jalur lokal akan memetakan perencanaan rute yang dipilih apakah rute tersebut benar dan dapat dieksekusi. Peta probabilistik terdiri dari pengambilan sampel jalur tanpa tabrakan pada perencanaan jalur lokal. Pada fase *query* adalah penggabungan rute perjalanan titik awal robot hingga titik tujuan robot. Algoritma *Improved A-star search (A*)* dalam perencanaan peta probabilistik yang dibentuk untuk mencari rute yang aman dari titik awal menuju titik tujuan. Algoritma *Improved A-star search (A*)* terdiri dari dua algoritma pemrosesan: tahap *pre-processing* dan tahap *post-processing*. Pada tahap *pre-processing* yaitu pencarian titik awal menuju titik tujuan dengan rute yang paling minimum dengan algoritma *A-star search (A*)*, untuk tahap *post-*

processing yaitu proses penyederhanaan atau normalisasi rute perjalanan yang telah didapat dalam proses *pre-processing* (Fu et al., 2018).

2.2.11 *Digital Differential Analyzer (DDA)*

Algoritma DDA digunakan untuk membuat garis lurus pada digital plotter. Cara kerja algoritma ini diawali dengan menentukan titik awal koordinat (x_1, y_1) dan titik akhir (x_2, y_2) . Selanjutnya, menghitung nilai dx dan dy yang direpresentasikan pada persamaan 2.5, dan 2.6.

$$dx = x_2 - x_1 \quad (2.5)$$

$$dy = y_2 - y_1 \quad (2.6)$$

dengan x_2 adalah posisi koordinat x target robot, x_1 posisi koordinat x robot saat ini, y_2 posisi koordinat y target robot, dan y_1 posisi koordinat y robot saat ini. Kemudian melakukan perhitungan jumlah langkah atau *step* untuk yang harus ditempuh agar sampai pada titik tujuan target. Apabila nilai dari $|dx| < |dy|$ maka jumlah langkah yang harus ditempuh terisi nilai $|dy|$, tetapi ketika nilai dari $|dx| > |dy|$ maka jumlah langkah yang harus ditempuh terisi nilai $|dx|$. Selanjutnya adalah menentukan perhitungan penambahan absis (x) dan ordinat (y) direpresentasikan dengan persamaan 2.7, dan 2.8.

$$x = \frac{dx}{step} \quad (2.7)$$

$$y = \frac{dy}{step} \quad (2.8)$$

Sehingga koordinat titik berikutnya (x_{i+1}, y_{i+1}) akan bernilai sama dengan koordinat titik saat ini yaitu (x_i, y_i) ditambah x dan y . Apabila nilai x dan y adalah bilangan bulat, maka akan dilakukan pembulatan. Proses perhitungan koordinat ini akan terus diulang hingga titik titik yang dihitung yaitu (x_{i+1}, y_{i+1}) mencapai titik akhir tujuan (x_2, y_2) (Shreiner, 2014).