

Perancangan Sistem Pintu Otomatis Menggunakan Heuristic Search Generate and Test

Putra Pratama Margiansyah^{*1}, Anang Kukuh Adisusilo², Nonot Wisnu Karyanto³

Abstrak

Perkembangan teknologi *Artificial Intelligence* mendorong terciptanya sistem otomatisasi yang mampu membantu aktivitas manusia secara lebih efektif dan efisien. Salah satu penerapannya adalah sistem pintu otomatis yang dapat membuka dan menutup pintu secara mandiri berdasarkan kondisi lingkungan. Permasalahan pada sistem konvensional adalah proses pengoperasian pintu masih dilakukan secara manual sehingga kurang praktis dan kurang responsif terhadap keberadaan pengguna. Penelitian ini bertujuan untuk merancang simulasi *Intelligent Agent* pada sistem pintu otomatis menggunakan metode *Generate and Test* sebagai bagian dari *Heuristic Search*. Sistem dirancang menggunakan bahasa pemrograman Python dengan antarmuka grafis berbasis *Tkinter*. Agent bekerja dengan menerima data jarak dari sensor sebagai *perception*, kemudian melakukan proses evaluasi jalur untuk menentukan tindakan membuka atau menutup pintu. Hasil penelitian menunjukkan bahwa jalur terbaik yang diperoleh adalah $S0 \rightarrow S1 \rightarrow S2 \rightarrow S3$ dengan total *cost* sebesar 3 karena berhasil mencapai *goal state*. Sistem juga mampu melakukan simulasi penanganan kesalahan seperti *sensor error*, *obstacle*, dan kesalahan mekanik. Berdasarkan hasil pengujian, sistem yang dirancang mampu memvisualisasikan proses pencarian solusi dan pengambilan keputusan secara interaktif serta berjalan sesuai dengan konsep *Intelligent Agent*.

Kata kunci: *Artificial Intelligence, Generate and Test, Heuristic Search, Intelligent Agent, Pintu Otomatis*

Abstract

The development of *Artificial Intelligence* technology has encouraged the creation of automation systems capable of assisting human activities more effectively and efficiently. One of its applications is an automatic door system that can open and close independently based on environmental conditions. The problem with conventional systems is that door operations are still performed manually, making them less practical and less responsive to user presence. This study aims to design an *Intelligent Agent* simulation for an automatic door system using the *Generate and Test* method as part of *Heuristic Search*. The system was developed using the Python programming language with a *Tkinter*-based graphical user interface. The agent receives distance data from sensors as *perception* input and then evaluates possible paths to determine the appropriate action for opening or closing the door. The results show that the best path obtained is $S0 \rightarrow S1 \rightarrow S2 \rightarrow S3$ with a total cost of 3 because it successfully reaches the goal state. The system is also capable of simulating error handling such as *sensor errors*, *obstacles*, and *mechanical failures*. Based on the testing results, the designed system successfully visualizes the solution-searching process and decision-making interactively in accordance with the *Intelligent Agent* concept.

Keywords: *Artificial Intelligence, Generate and Test, Heuristic Search, Intelligent Agent, Smart Door System*

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi *Artificial Intelligence* (AI) telah mendorong terciptanya berbagai sistem otomatisasi yang mampu membantu aktivitas manusia secara lebih efektif dan efisien. Salah satu implementasi teknologi tersebut adalah sistem pintu otomatis yang dapat membuka dan menutup pintu secara mandiri berdasarkan kondisi lingkungan. Sistem pintu otomatis banyak digunakan pada pusat perbelanjaan, rumah sakit, dan gedung perkantoran karena dinilai lebih praktis dan mampu meningkatkan kenyamanan pengguna[4].

Sistem pintu otomatis bekerja dengan memanfaatkan sensor untuk mendeteksi keberadaan manusia sehingga pintu dapat terbuka secara otomatis tanpa kontak langsung. Selain meningkatkan efisiensi, sistem otomatis juga dapat memberikan respons yang lebih cepat terhadap kondisi lingkungan sekitar.

1.2 Permasalahan

Pada sistem konvensional, proses membuka dan menutup pintu masih dilakukan secara manual sehingga kurang efisien, terutama pada lingkungan dengan mobilitas pengguna yang tinggi. Selain itu,

sistem manual tidak mampu memberikan respons otomatis terhadap keberadaan pengguna maupun kondisi tertentu pada lingkungan sekitar.

Beberapa penelitian sebelumnya juga masih berfokus pada implementasi perangkat keras tanpa menyediakan simulasi visual mengenai proses pengambilan keputusan agent dan pencarian solusi secara interaktif. Oleh karena itu, diperlukan suatu sistem yang mampu memvisualisasikan proses kerja agent cerdas pada sistem pintu otomatis.

1.3 Penelitian Terkait

Penelitian oleh Maulana [1] menjelaskan bahwa metode Generate and Test digunakan untuk menghasilkan beberapa kemungkinan solusi kemudian melakukan evaluasi guna memperoleh solusi terbaik. Penelitian lain yang dilakukan oleh Welianto et al. [2] menunjukkan bahwa metode Generate and Test efektif digunakan dalam proses pencarian jalur karena mampu mengevaluasi beberapa kemungkinan keadaan sebelum menentukan hasil akhir.

Selain itu, penelitian oleh Gall et al. [3] menjelaskan bahwa Heuristic Search dapat membantu sistem dalam menentukan jalur optimal menuju goal state berdasarkan nilai heuristik pada setiap state. Penelitian Sharma dan Bodkhe [4] membahas sistem pintu otomatis menggunakan sensor dan microcontroller untuk mendeteksi keberadaan manusia sehingga pintu dapat terbuka secara otomatis.

1.4 Kebaruan Penelitian (Novelty)

Berbeda dengan penelitian sebelumnya, penelitian ini tidak hanya berfokus pada sistem otomatisasi pintu, tetapi juga menambahkan visualisasi state space graph, simulasi jalur pencarian solusi, animasi pembukaan pintu, serta penanganan kesalahan (error handling) pada sistem berbasis Intelligent Agent. Sistem dirancang menggunakan bahasa pemrograman Python dengan antarmuka grafis berbasis Tkinter sehingga proses kerja agent dapat divisualisasikan secara lebih interaktif dan mudah dipahami.

II. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode Heuristic Search dengan pendekatan Generate and Test untuk menentukan jalur terbaik pada sistem agent cerdas pintu otomatis. Metode Generate and Test bekerja dengan menghasilkan beberapa kemungkinan solusi (generate), kemudian melakukan evaluasi terhadap setiap solusi (test) untuk memperoleh jalur yang valid menuju goal state [1], [2], .

Sistem dirancang menggunakan bahasa pemrograman Python dengan antarmuka grafis berbasis Tkinter. Selain itu, penelitian ini juga menerapkan konsep Intelligent Agent yang mampu menerima informasi dari lingkungan melalui sensor, memproses kondisi yang diterima, dan menentukan aksi secara otomatis [4].

2.1 Perancangan Intelligent Agent

Perancangan agent pada penelitian ini menggunakan konsep PAGE (Perception, Action, Goal, Environment). Konsep tersebut digunakan untuk mendefinisikan cara kerja agent dalam sistem pintu otomatis.

1. Perception (P)

Agent menerima informasi dari sensor untuk mengetahui kondisi lingkungan di sekitar pintu, yaitu:

- Ada orang di depan pintu
- Tidak ada orang di depan pintu
- Jarak orang terhadap pintu

2. Action (A)

Berdasarkan kondisi yang diterima dari sensor, agent dapat melakukan beberapa aksi, yaitu:

- Membuka pintu
- Menutup pintu

3. Goal (G)

Tujuan utama dari sistem adalah:

- Membuka pintu secara otomatis saat terdapat orang

- Menutup pintu ketika tidak ada orang
- Meningkatkan kenyamanan pengguna
- Mengurangi kontak fisik dengan pintu

4. Environment (E)

Lingkungan kerja agent meliputi:

- Area pintu masuk
- Pengguna yang mendekati pintu
- Kondisi sekitar pintu otomatis

Konsep Intelligent Agent digunakan agar sistem mampu mengambil keputusan secara otomatis berdasarkan kondisi lingkungan [3], [4].

2.2 Action Function

Action Function digunakan untuk menentukan aksi yang dilakukan agent berdasarkan hasil pembacaan sensor.

Perception	Action
Ada Orang	Buka Pintu
Tidak Ada Orang	Tutup Pintu

2.3 Cara Kerja Sistem

Sistem bekerja dengan memanfaatkan sensor jarak untuk mendeteksi keberadaan pengguna di depan pintu. Sensor akan mengirimkan data jarak ke agent sebagai input. Selanjutnya, agent akan mengevaluasi kondisi berdasarkan nilai jarak yang diterima.

Apabila jarak pengguna kurang dari atau sama dengan 100 cm, maka sistem menganggap terdapat orang yang mendekati pintu sehingga pintu akan terbuka secara otomatis. Sebaliknya, apabila jarak lebih dari 100 cm, maka sistem akan menutup pintu.

Proses tersebut dilakukan secara berulang (looping) sehingga sistem mampu bekerja secara real-time [4].

2.4 Formulasi Problem

a. Initial State

Keadaan awal sistem adalah:

- Pintu dalam kondisi tertutup
- Tidak terdapat orang di depan pintu
- Sensor aktif membaca jarak

b. Successor Function

Aksi yang dilakukan agent berdasarkan kondisi lingkungan ditunjukkan pada Tabel berikut.

Kondisi	Aksi
Jarak ≤ 100 cm	Buka pintu
Jarak > 100 cm	Tutup pintu

c. Goal Test

Kondisi goal tercapai apabila:

- Pintu terbuka saat terdapat orang
- Pintu tertutup saat tidak terdapat orang

d. Path Cost

Setiap perpindahan state memiliki nilai cost. Contoh perhitungan jalur valid:

$$1+1+1=3$$

Jalur :

$$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$$

Total cost : 3

2.5 State Space Graph

Pada penelitian ini digunakan state space graph untuk menggambarkan perpindahan kondisi pada sistem pintu otomatis.

Keterangan state:

- S0 : Pintu tertutup (initial state)
- S1 : Orang terdeteksi
- S2 : Pintu terbuka
- S3 : Goal state
- E1 : Sensor error
- I1 : Idle state
- O1 : Obstacle state

Metode Heuristic Search digunakan untuk menentukan jalur terbaik menuju goal state berdasarkan nilai heuristik dari setiap state [3] .

2.6 Implementasi Heuristic Search

Penelitian ini menerapkan beberapa metode Heuristic Search, yaitu:

- Generate and Test
- Hill Climbing
- Greedy Algorithm
- A* Algorithm

a. Generate and Test

Metode ini menghasilkan beberapa kemungkinan jalur kemudian melakukan evaluasi untuk memperoleh solusi terbaik [1], [2] .

No	Jalur	Valid	Total Cost
1	S0 → S1 → S2 → S3	Ya	3
2	S0 → S1 → E1	Tidak	2
3	S0 → I1	Tidak	1
4	S0 → S1 → S2 → O1	Tidak	4

Jalur terbaik:

S0 → S1 → S2 → S3

Dengan Total cost Sebesar

1+1=3

b. Hill Climbing

Metode Hill Climbing merupakan salah satu metode Heuristic Search yang bekerja dengan memilih state terbaik berdasarkan nilai heuristik terkecil pada setiap langkah pencarian [3]. Metode ini melakukan pencarian secara bertahap dari initial state menuju goal state dengan cara membandingkan seluruh tetangga (neighbor node) yang tersedia, kemudian memilih node yang memiliki nilai heuristik paling baik.

Pada sistem pintu otomatis, metode Hill Climbing digunakan untuk menentukan jalur tercepat menuju kondisi tujuan (goal state), yaitu keadaan ketika pengguna berhasil melewati pintu otomatis.

Nilai heuristik yang digunakan pada penelitian ini ditunjukkan sebagai berikut:

State	Keterangan	h(n)
S0	Pintu tertutup	3
S1	Orang terdeteksi	2
S2	Pintu terbuka	1
S3	Goal state	0
E1	Sensor error	3
O1	Obstacle	2
I1	Idle	4

Semakin kecil nilai heuristik, maka state tersebut dianggap semakin dekat dengan goal state.

c. Greedy Algorithm

Metode Greedy Algorithm merupakan metode pencarian yang memilih node dengan nilai heuristik paling kecil tanpa mempertimbangkan keseluruhan jalur yang telah dilalui [3]. Metode ini berfokus pada pencarian solusi tercepat menuju goal state berdasarkan kondisi terbaik saat itu (best immediate choice).

Pada sistem pintu otomatis, metode Greedy Algorithm digunakan untuk menentukan jalur pembukaan pintu secara cepat berdasarkan nilai heuristik terdekat menuju kondisi tujuan.

Nilai Heuristik

State	$h(n)$
S0	3
S1	2
S2	1
S3	0
E1	3
O1	2

d. Tree Terminology

Dalam penelitian ini, representasi masalah juga dapat dijelaskan menggunakan konsep tree search untuk mempermudah pemahaman alur pencarian solusi dari initial state menuju goal state. Struktur tree menggambarkan hubungan antar node berdasarkan proses ekspansi dari setiap kondisi sistem.

1. Root Node

Root node merupakan titik awal dari pencarian, yaitu kondisi awal sistem sebelum dilakukan proses apa pun.

- S0 : Pintu dalam kondisi tertutup (initial state)

2. Parent Node dan Child Node

- Parent node adalah node yang menjadi asal dari percabangan.
 - Child node adalah node hasil ekspansi dari parent node.
- Contoh:
- S0 adalah parent dari S1 dan I1
 - S1 adalah parent dari S2 dan E1
 - S2 adalah parent dari S3 dan O1

3. Branch (Cabang)

Branch menunjukkan jalur kemungkinan yang dapat diambil oleh agent dari satu state ke state lainnya.

Contoh cabang:

S0 → S1 → S2 → S3

S0 → I1 (dead-end)

S1 → E1 (error state)

S2 → O1 (obstacle)

4. Leaf Node

Leaf node adalah node terakhir yang tidak memiliki cabang lanjutan.

Pada sistem ini, contoh leaf node:

- S3 (Goal state)
- E1 (Sensor error)
- O1 (Obstacle)
- I1 (Idle state)

5. Dead State (State Gagal)

Contoh:

- S0 → I1 (tidak ada kelanjutan)
- S1 → E1 (sensor error)
- S2 → O1 (terhalang obstacle)

III. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Sistem

Pada penelitian ini telah berhasil dirancang sebuah sistem simulasi pintu otomatis berbasis Intelligent Agent menggunakan metode Heuristic Search Generate and Test. Sistem dikembangkan menggunakan bahasa pemrograman Python dengan antarmuka grafis berbasis Tkinter sehingga mampu menampilkan proses kerja agent secara visual dan interaktif.

Sistem bekerja dengan menerima data masukan (perception) berupa jarak pengguna terhadap pintu. Data tersebut kemudian diproses oleh agent untuk menentukan aksi yang sesuai, yaitu membuka atau menutup pintu secara otomatis. Apabila sensor mendeteksi keberadaan pengguna dengan jarak kurang dari atau sama dengan 100 cm, maka agent akan memberikan perintah membuka pintu. Sebaliknya, apabila tidak terdapat pengguna di depan pintu atau jarak lebih dari 100 cm, maka pintu akan tertutup secara otomatis [4].

Implementasi sistem dilakukan dengan beberapa komponen utama, yaitu:

- Sensor simulasi sebagai input perception
- Agent cerdas sebagai pengambil keputusan
- Visualisasi state space graph
- Simulasi pembukaan dan penutupan pintu
- Penanganan kondisi kesalahan (error handling)

Sistem juga dirancang menggunakan konsep PAGE (Perception, Action, Goal, Environment) sehingga mampu bekerja sesuai dengan konsep Intelligent Agent. Agent dapat menerima informasi dari lingkungan, melakukan evaluasi kondisi, kemudian menentukan aksi secara otomatis [3].

3.2 Hasil Pengujian Generate and Test

Metode Generate and Test digunakan untuk menghasilkan beberapa kemungkinan jalur (generate) kemudian melakukan evaluasi terhadap setiap jalur (test) untuk memperoleh solusi terbaik [1], [2]. Pada penelitian ini dilakukan pengujian terhadap beberapa kemungkinan kondisi yang dapat terjadi pada sistem pintu otomatis.

Hasil pengujian metode Generate and Test

No	Jalur	Status	Total Cost
1	S0 → S1 → S2 → S3	Valid	3
2	S0 → S1 → E1	Tidak Valid	2
3	S0 → I1	Tidak Valid	1
4	S0 → S1 → S2 → O1	Tidak Valid	4
5	S0 → S1 → S2	Belum Goal	2

Berdasarkan hasil pengujian tersebut diperoleh bahwa jalur terbaik adalah:

S0 → S1 → S2 → S3

Keterangan state:

- S0 : Pintu tertutup (initial state)
- S1 : Orang terdeteksi
- S2 : Pintu terbuka
- S3 : Goal state
- E1 : Sensor error
- O1 : Obstacle
- I1 : Idle state

Perhitungan path cost dilakukan dengan memberikan nilai cost sebesar 1 pada setiap perpindahan state.

Perhitungan total cost:

$1+1+1=3$

Keterangan:

- S0 → S1 = 1
- S1 → S2 = 1

- $S2 \rightarrow S3 = 1$

Total cost yang diperoleh adalah 3 sehingga jalur tersebut dinyatakan sebagai jalur paling optimal karena berhasil mencapai goal state. Jalur lain tidak dipilih karena mengalami kondisi error, obstacle, atau tidak mencapai tujuan sistem.

Hasil tersebut menunjukkan bahwa metode Generate and Test mampu melakukan evaluasi terhadap beberapa kemungkinan solusi dan memilih jalur terbaik berdasarkan validitas jalur serta nilai cost yang dimiliki [1].

3.3 Analisis State Space Graph

Pada penelitian ini digunakan state space graph untuk menggambarkan proses perpindahan kondisi pada sistem pintu otomatis. Penggunaan state space graph bertujuan untuk mempermudah proses visualisasi pencarian solusi yang dilakukan oleh agent.

State awal dimulai dari kondisi S0 ketika pintu dalam keadaan tertutup. Selanjutnya sistem akan berpindah menuju S1 apabila sensor mendeteksi keberadaan pengguna. Setelah itu sistem menuju S2 ketika pintu terbuka secara otomatis, kemudian mencapai S3 sebagai goal state ketika pengguna berhasil melewati pintu.

Selain jalur utama, sistem juga memiliki beberapa kemungkinan state lain seperti:

- E1 (sensor error)
- O1 (obstacle detection)
- I1 (idle state)

Keberadaan state tambahan tersebut membuat sistem menjadi lebih realistis karena mampu menggambarkan kondisi yang mungkin terjadi pada sistem nyata. Metode Heuristic Search digunakan untuk membantu agent menentukan jalur terbaik menuju goal state berdasarkan nilai heuristik dari setiap state [3]. Semakin kecil nilai heuristik suatu node, maka node tersebut dianggap semakin dekat dengan tujuan.

3.4 Hasil Pengujian Hill Climbing

Metode Hill Climbing digunakan untuk menentukan jalur terbaik dengan memilih node yang memiliki nilai heuristik paling kecil pada setiap langkah pencarian [3].

Nilai heuristik yang digunakan pada penelitian pada tabel 2 :

State	Keterangan	$h(n)$	State
S0	Pintu tertutup	3	S0
S1	Orang terdeteksi	2	S1
S2	Pintu terbuka	1	S2
S3	Goal state	0	S3
E1	Sensor error	3	E1
O1	Obstacle	2	O1

Proses pencarian metode Hill Climbing dilakukan sebagai berikut:

- Dari S0 dipilih S1 karena memiliki heuristik lebih kecil dibanding I1
- Dari S1 dipilih S2 karena memiliki heuristik lebih kecil dibanding E1
- Dari S2 dipilih S3 karena merupakan goal state

Jalur yang diperoleh:

$S0 \rightarrow S1 \rightarrow S2 \rightarrow S3$

Total cost:

$1+1+1=3$

Hasil pengujian menunjukkan bahwa metode Hill Climbing berhasil mencapai goal state dengan memilih node terbaik pada setiap langkah pencarian.

3.5 Hasil Pengujian Greedy Algorithm

Metode Greedy Algorithm bekerja dengan memilih node yang memiliki nilai heuristik paling kecil tanpa mempertimbangkan keseluruhan jalur yang telah dilalui [3].

Pada penelitian ini proses pencarian dilakukan sebagai berikut:

- Dari S0 memilih S1
- Dari S1 memilih S2

- Dari S2 memilih S3

Jalur yang diperoleh:

$S0 \rightarrow S1 \rightarrow S2 \rightarrow S3$

Total cost:

$1+1+1=3$

Hasil pengujian menunjukkan bahwa metode Greedy Algorithm mampu menemukan jalur menuju goal state secara cepat karena selalu memilih node dengan nilai heuristik terkecil.

Namun demikian, metode ini memiliki kelemahan karena hanya berfokus pada solusi terbaik saat itu (local optimum) tanpa mempertimbangkan kemungkinan jalur lain yang mungkin lebih optimal secara keseluruhan.

3.6 Hasil Pengujian A* Algorithm

Metode A* Algorithm digunakan untuk menentukan jalur terbaik berdasarkan kombinasi nilai cost perjalanan dan nilai heuristik [3].

Persamaan evaluasi node pada metode A* adalah:

$$f(n) = g(n) + h(n)$$

Keterangan:

- $g(n)$ = biaya perjalanan dari state awal
- $h(n)$ = estimasi jarak menuju goal
- $f(n)$ = total evaluasi node

Hasil perhitungan metode A* ditunjukkan pada Tabel 3:

Node	$g(n)$	$h(n)$	$f(n)$
S0	0	3	3
S1	1	2	3
S2	2	1	3
S3	3	0	3

Proses pencarian:

- Dari S0 menuju S1
- Dari S1 menuju S2
- Dari S2 menuju S3

Jalur yang diperoleh:

$S0 \rightarrow S1 \rightarrow S2 \rightarrow S3$

Total cost:

3

Hasil pengujian menunjukkan bahwa metode A* Algorithm berhasil menemukan jalur optimal menuju goal state dengan mempertimbangkan cost perjalanan dan nilai heuristik secara bersamaan.

3.7 Pengujian Sistem Secara Keseluruhan

Pengujian sistem dilakukan untuk mengetahui kemampuan agent dalam merespons berbagai kondisi lingkungan secara otomatis.

Tabel 4. Hasil Pengujian Sistem

Kondisi	Respons Sistem	Status	Kondisi
Orang terdeteksi	Pintu terbuka	Berhasil	Orang terdeteksi
Tidak ada orang	Pintu tertutup	Berhasil	Tidak ada orang
Sensor error	Sistem menampilkan error	Berhasil	Sensor error
Obstacle terdeteksi	Pintu tetap terbuka	Berhasil	Obstacle terdeteksi
Kondisi idle	Sistem standby	Berhasil	Kondisi idle

Berdasarkan hasil pengujian, sistem mampu bekerja sesuai dengan konsep Intelligent Agent. Agent berhasil menerima data dari lingkungan, memproses kondisi, kemudian menentukan aksi secara otomatis tanpa campur tangan manusia.

Sistem juga mampu melakukan simulasi kondisi kesalahan seperti sensor error dan obstacle detection. Hal tersebut menunjukkan bahwa sistem yang dirancang memiliki kemampuan simulasi yang cukup baik untuk menggambarkan proses kerja pintu otomatis secara interaktif dan real-time [4].

3.8 Pembahasan

Berdasarkan seluruh hasil pengujian yang telah dilakukan, metode Generate and Test terbukti efektif digunakan dalam simulasi sistem pintu otomatis berbasis Intelligent Agent. Metode ini mampu menghasilkan beberapa kemungkinan solusi kemudian melakukan evaluasi untuk menentukan jalur terbaik menuju goal state [1], [2].

Selain itu, penerapan metode Heuristic Search juga membantu agent dalam menentukan keputusan secara lebih terarah berdasarkan nilai heuristik dari setiap state [3]. Penggunaan visualisasi state space graph dan antarmuka grafis berbasis Tkinter membuat proses simulasi menjadi lebih interaktif dan mudah dipahami oleh pengguna.

Hasil penelitian menunjukkan bahwa seluruh metode pencarian yang digunakan menghasilkan jalur optimal yang sama, yaitu:

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$

dengan total cost sebesar: 3

Hal tersebut menunjukkan bahwa sistem mampu mencapai goal state secara optimal sesuai tujuan penelitian. Namun demikian, penelitian ini masih memiliki beberapa keterbatasan, yaitu sistem masih berupa simulasi dan belum diimplementasikan secara langsung pada perangkat keras (hardware) nyata. Selain itu, sensor yang digunakan masih berupa simulasi input sehingga belum mampu mengukur kondisi lingkungan sebenarnya secara langsung.

Meskipun demikian, sistem yang dirancang telah berhasil memvisualisasikan proses pengambilan keputusan agent secara otomatis dan dapat dijadikan dasar pengembangan sistem pintu otomatis berbasis Artificial Intelligence pada penelitian selanjutnya.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem pintu otomatis berbasis Intelligent Agent menggunakan metode Heuristic Search Generate and Test berhasil dirancang dan diimplementasikan dengan baik menggunakan bahasa pemrograman Python berbasis Tkinter. Sistem mampu menerima data dari lingkungan melalui sensor sebagai perception, kemudian melakukan proses evaluasi kondisi untuk menentukan aksi membuka atau menutup pintu secara otomatis.

Hasil pengujian menunjukkan bahwa metode Generate and Test mampu menghasilkan beberapa kemungkinan jalur dan memilih solusi terbaik menuju goal state. Jalur optimal yang diperoleh pada penelitian ini adalah:

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$

dengan total cost sebesar:

$1+1+1=3$

Selain metode Generate and Test, penelitian ini juga menerapkan metode Hill Climbing, Greedy Algorithm, dan A* Algorithm. Seluruh metode berhasil mencapai goal state dengan jalur optimal yang sama. Hal tersebut menunjukkan bahwa penerapan metode Heuristic Search dapat membantu sistem dalam menentukan keputusan secara otomatis berdasarkan nilai heuristik setiap state.

Sistem yang dirancang juga mampu memvisualisasikan proses pencarian solusi, perpindahan state, serta simulasi kondisi kesalahan seperti sensor error, idle state, dan obstacle detection secara interaktif. Dengan demikian, tujuan penelitian untuk merancang simulasi sistem pintu otomatis berbasis Intelligent Agent telah berhasil tercapai sesuai dengan hasil pengujian dan pembahasan yang dilakukan.

Penelitian ini masih memiliki beberapa keterbatasan karena sistem yang dibuat masih berupa simulasi dan belum diterapkan langsung pada perangkat keras (hardware) nyata. Oleh karena itu, penelitian selanjutnya dapat dikembangkan dengan mengintegrasikan sensor asli seperti sensor ultrasonik atau infrared, menggunakan microcontroller seperti Arduino atau ESP32, serta menambahkan teknologi Internet of Things (IoT) agar sistem dapat bekerja secara lebih optimal dan real-time pada lingkungan sebenarnya.

REFERENSI

- [1] A. Maulana, “Pemanfaatan Algoritma Generate and Test Dalam Kasus Travelling Salesman Problem Pencarian Jalur Terpendek,” *json*, vol. 3, no. 2, p. 128, Dec. 2021, doi: 10.30865/json.v3i2.3617.
- [2] S. Welianto, R. G. Santosa, and A. R. Chrismanto, “IMPLEMENTASI ALGORITMA GENERATE AND TEST PADA PENCARIAN RUTE TERPENDEK,” *INF*, vol. 7, no. 2, Jul. 2012, doi: 10.21460/inf.2011.72.103.
- [3] Lymmy Ogbidi and Benneth Oteh, “Advances in Hybrid Machine Learning and Physics-Based Models for Enhanced Reservoir Simulation,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol*, vol. 10, no. 6, pp. 2533–2543, Dec. 2024, doi: 10.32628/IJSRCSEIT.
- [4] A. Lissovoi, P. Oliveto, and J. A. Warwicker, “How the Duration of the Learning Period Affects the Performance of Random Gradient Selection Hyper-Heuristics,” *AAAI*, vol. 34, no. 03, pp. 2376–2383, Apr. 2020, doi: 10.1609/aaai.v34i03.5617.