

REVIEW ALGORITMA C-THETA* DAN PROSPEK IMPLEMENTASINYA DI MASA DEPAN

Nathanael Aldo Phillip*¹, Ketut Bayu Yoga Bintoro*², Silvester Dian Handy Permana*³

*Universitas Trilogi

Program Studi Teknik Informatika,

Jl. Kampus Trilogi / STEKPI No.1, Kalibata, Jakarta Selatan, 12760. Indonesia

Email : nathanaelphillip@trilogi.ac.id*¹, ketutbayu@trilogi.ac.id*², handy@trilogi.ac.id*³

Abstrak

Perencanaan jalan digunakan untuk menentukan jalan yang akan dilalui oleh agen agar dapat mencapai tujuannya. Penggunaan perencanaan jalan tidak hanya diterapkan pada game melainkan juga diterapkan pada robot dan mesin lainnya yang memiliki kemampuan untuk bergerak. C-Theta* merupakan algoritma perencanaan jalan baru dengan kecepatan serta hasil perencanaan yang lebih baik dibandingkan dengan pendahulunya, algoritma A* dan Theta*. Tentunya sebagai algoritma yang baru C-Theta* belum memiliki penelitian lebih lanjut mengenai prospek implementasinya di masa depan, sehingga para peneliti tidak dapat mengetahui apakah algoritma ini dapat diimplementasikan pada penelitiannya. Penelitian ini bertujuan untuk memberi review mengenai C-Theta* dengan menggunakan metode komparatif, yaitu membandingkan C-Theta* dengan algoritma perencanaan jalan lainnya dan menganalisa kasus pada penelitian lain. Guna mencari tahu kemungkinan bagi implementasi algoritma ini pada kasus-kasus yang diuji. Hasil dari penelitian ini membuktikan bahwa C-Theta* memiliki kesamaan dengan grid-based lainnya seperti A* dan Theta*, sehingga memungkinkan bagi C-Theta* untuk menggantikan perencanaan jalan dengan algoritma A* dan algoritma grid-based lainnya.

Keywords: Perencanaan Jalan, C-Theta*, Pengujian, Implementasi dimasa depan, Kecerdasan Buatan

Abstract

Path planning is used to determine the path the agent will travel through in order to achieve its objectives. The use of path planning is not only applicable to robots and other machines that have the ability to move. C-Theta * is a new path planning algorithm with speed and better planning results compared to its predecessor, the A * and Theta * algorithms. Of course, as the new algorithm C-Theta * has no further research on the prospects for future implementation, so researchers cannot know whether this algorithm can be implemented in the research. This study aims to provide a review of C-Theta * by using the comparative method, for comparing C-Theta * with a path planning algorithm and analyzing cases in other studies. To find out the possibility of implementing this algorithm in the test cases. The results of this study prove that C-Theta * has similarities with grid-based as A * and Theta *, making it possible for C-Theta * to work on A * algorithms and other grid-based algorithms.

Keywords : *path-planning, C-Theta*, review, future implementation, Artificial Intelligence*

I. PENDAHULUAN

Perencanaan jalan (dalam bahasa Inggris disebut *path-planning*) adalah aplikasi penting yang diperlukan oleh agen pada bidang Artificial Intelligence. Penggunaan perencanaan jalan tidak hanya diterapkan pada game melainkan juga diterapkan pada robot dan mesin lainnya yang memiliki kemampuan untuk bergerak. Perencanaan jalan memiliki pendekatan dasar yang terdiri dari 3 kunci utama. Ke-3 kunci ini adalah representasi graph dari map yang ada, algoritma pencarian, dan fungsi heuristic untuk mengarahkan pencariannya. Dalam penelitian ini akan dibahas salah satu dari ketiga kunci di atas yaitu algoritma pencarian. Algoritma yang digunakan dalam perencanaan jalan menentukan jalan yang akan dilalui oleh agen. Apabila implementasi algoritma tidak cocok dengan bidang yang diimplementasikan, maka akan mengakibatkan algoritma tidak dapat memberikan jalan yang terbaik untuk agen tersebut. [1]

C-Theta* merupakan algoritma perencanaan jalan terbaru dengan hasil pencarian jalan yang lebih baik bila dibandingkan dengan algoritma pendahulunya yaitu Theta*. C-Theta* juga memiliki kecepatan perencanaan yang lebih cepat dibandingkan dengan algoritma A*. Hal ini dikarenakan C-Theta* merupakan gabungan dari Theta* dan A*. Tentunya sebagai algoritma yang baru C-Theta* belum memiliki penelitian lebih lanjut mengenai prospek implementasinya di masa depan, sehingga para peneliti tidak dapat mengetahui apakah algoritma ini dapat diimplementasikan pada penelitiannya. Demi mengembangkan algoritma perencanaan jalan di berbagai bidang maka perlu dilakukan penelitian lebih lanjut untuk mencari tahu prospek

implementasi dari algoritma C-Theta* di masa depan.

Penelitian ini bertujuan untuk memberi review algoritma C-Theta* melalui analisa kecocokan algoritma C-Theta* dengan implementasinya. Guna mencari tahu kemungkinan bagi implementasi algoritma ini pada kasus-kasus yang diuji. Diharapkan melalui hasil review ini para ahli dan peneliti dapat mengembangkan penelitian mereka dengan menggunakan algoritma C-Theta*.

II. LANDASAN TEORI

Algoritma A*

A* merupakan algoritma yang seringkali digunakan untuk mencari solusi dalam memecahkan masalah-masalah yang ada, salah satunya adalah masalah pencarian jalan. Dapat dikatakan bahwa A* adalah algoritma yang paling populer untuk perencanaan jalan pada bidang games dan robot [2]. Sebagai perkembangan dari algoritma *Dijkstra*, A* banyak digunakan pada game dengan genre strategi yang menggunakan grids dalam proses perencanaan jalannya [3].

A* memiliki sejumlah varian yang telah dikembangkan demi menyelesaikan isu dimana A* tidak dapat menggunakan *any angle pathfinding* (pencarian jalan melalui segala sudut) dengan melakukan pengecekan *line of sight* untuk menentukan jalan di depannya. Tidak seperti A* yang menganggap pusat dari setiap tiles dari sebuah grid, dalam *any angle pathfinding* setiap sudut pada grid dipertimbangkan sebagai ekspansi node dari titik asal ke tujuan. A* mengevaluasi setiap node untuk

mencari langkah terbaik ke tujuan yang memiliki perhitungan jarak tercepat [4].

```

Main()
S(start) := 0; (Start Node)
parent := S(start);
open := 0; /*(Open list := Set of node under consideration)*/
f(S(start)) := g(S(start)) + h(S(start)); /*(g(s) := Actual cost, h(s) := heuristic cost,
f(s) = Total estimated cost to reach goal) */

open.Insert(S(start), f(S(start)));
closed := 0; /*( Closed List := Set of nodes already evaluated) */
while open != 0 do
  S := open.pop();
  if S == S(goal) then
    return "Path Found";
  closed := closed U {S}
  for each S' ∈ neighbours (s) do
    if S' ∉ closed then
      if S' ∉ open then
        g(S') := ∞;
        parent(S') := NULL;
        NodeValue(S, S');
      return "path not found";
end;
NodeValue(s, s')
if g(s) + c(s,s') < g(s') then /*(where c(s,s') cost to move from s to successor node s')*/
  g(s') := g(s) + c(s,s');
  parent(s') := s;
  if s' ∈ open then
    open.pop(s');
  open.push(s', f(s'));
end;
  
```

Gambar 1. Pseudocode A* [4]

Algoritma Theta*

Theta* adalah algoritma pencarian any-angle yang berarti pencarian segala sudut. Secara empiris Theta* dapat mencari jalan yang lebih pendek bila dibandingkan dengan algoritma A*. Hal ini dikarenakan Theta* menggabungkan kemampuan A* pada grafik visibilitas (dimana perubahan arah terjadi hanya pada sudut-sudut dari sel yang terhalang) dan kemampuan A* pada grids (dimana setiap sudut berkembang sejumlah dengan sel yang ada). Walaupun demikian, algoritma Theta* memiliki kelemahan dalam proses pencarian yang lama. Perbedaan utama antara Theta* dan A* pada grid adalah parent dari sebuah vertex dapat berupa sudut manapun pada Theta*, sedangkan parent dari sebuah vertex pada A* harus berdekatan dengan vertex tersebut [5].

```

NodeValue(s,s')
If LOS (parent(s),s') then
/* Decision - 1 */
  If(g(parent(s))+c(parent(s),s')) < g(s') then
    g(s') := g(parent(s)) + c(parent(s),s');
    parent (s') = parent(s);
    if s' ∈ open then
      open.pop(s');
    open.push(s', f(s'))
  else /*Decision -2 */
    If g(s) + c(s,s') < g(s') then /*(where c(s,s') cost to move from s to successor node s')*/
      g(s') := g(s) + c(s,s');
      parent (s') := s;
      if s' ∈ open then
        open.pop(s');
        open.push(s',f(s'));
end;
  
```

Gambar 2. Pseudocode Theta* [4].

Algoritma C-Theta*

C-Theta* merupakan algoritma perencanaan jalan baru yang dibuat dan diteliti oleh Mendonca, 2015. Algoritma ini memisahkan peta menjadi beberapa daerah dan dengan menggunakan area yang memiliki halangan sebagai bantuan bagi algoritma agar dapat memutuskan daerah mana yang diinginkan untuk pengecekan *line of sight* dan mana yang tidak. Cara ini dilakukan dengan memberikan sebuah label yang memiliki kepadatan tinggi dan rendah ke area tertentu berdasarkan jumlah dari halangan yang ada di area tersebut. Jika sebuah area memiliki label dengan kepadatan rendah maka algoritma yang digunakan adalah A* dan jika sebaliknya maka algoritma Theta* akan digunakan [4].

```

Main()
ConvertToRegions
do K-Means Clustering & Label Regions
/*start*/
S(start) := 0; (Start Node)
parent := S(start);
open := 0; /*(Open list := Set of node under consideration)*/
f(S(start)) := g(S(start)) + h(S(start)); /*(g(s) := Actual cost, h(s) := heuristic cost,
f(s) = Total estimated cost to
reach goal) */

open.Insert(S(start), f(S(start)));
closed := 0; /*( Closed List := Set of nodes already evaluated) */
while open != 0 do
  S := open.pop();
  If S == S(goal) then
    return "Path Found";
closed := closed U {S}
for each S' ∈ neighbours (s) do
  if S' ∉ closed then
    if S' ∉ open then
      g(S') := ∞;
      parent(S') := NULL;
      NodeValue(S, S');
return "path not found";
end;
NodeValue(s, s')
If DetermineNodeRegion(s') then *Low density Region*/
If LOS (parent(s), s') then
  If (g(parent(s)) + c(parent(s), s')) < g(s') then
    g(s') := g(parent(s)) + c(parent(s), s');
    parent(s') = parent(s);
    if s' ∈ open then
      open.pop(s');
    open.push(s', f(s'))
else
  If g(s) + c(s, s') < g(s') then /*(where c(s, s') cost to move
from s to successor node s')*/
    g(s') := g(s) + c(s, s');
    parent(s') := s;
    if s' ∈ open then
      open.pop(s');
    open.push(s', f(s'));
end

```

Gambar 3. Pseudocode C-Theta* [4].

III. REVIEW DAN PROSPEK

3.1 Metode Penelitian

Metode penelitian yang akan digunakan adalah metode komparatif, yaitu dengan melakukan review singkat pada setiap kasus dan mencari syarat-syarat yang dibutuhkan untuk implementasi dan prospek di masa depan. Metode komparatif ini mengganti metode yang sudah ada sebelumnya dengan metode C-Theta*. Apabila syarat-syarat ini dapat dipenuhi oleh C-Theta* maka tema yang sesuai akan diajukan untuk penelitian di masa depan [6].

3.2 Penggunaan C-Theta* pada Lingkungan dengan Kondisi Statis

3.2.1. *Grids*. Pada dasarnya C-Theta* memang dirancang untuk diimplementasikan pada lingkungan yang statis seperti Grid Map, dimana pada Grid Map setiap node terhubung satu sama lain layaknya jaring dengan berbagai bentuk. C-Theta* sendiri belum diuji-coba pada lingkungan yang bersifat dinamis. Namun, pembuktian hasil uji dari C-Theta* pada lingkungan statis terbukti memiliki hasil kinerja yang lebih baik daripada A* ataupun Theta*. Hal ini sudah dibuktikan oleh [4].

Pada tahun 2013 [7] melakukan penelitian mengenai perencanaan jalan pada grids dengan 8 titik neighbour yang lebih optimal dengan menggunakan tujuan pengganti sementara (Subgoal). Dengan menyisipkan satu atau beberapa subgoal sepanjang perjalanan dari titik start ke titik tujuan untuk mempermudah proses pencarian dari algoritma yang diterapkan. Peneliti menggunakan algoritma A* dalam pengujian subgoal di grid yang dibuat oleh peneliti. Metode ini juga dapat diterapkan pada C-Theta* karena masih berbasis grid.

3.2.1.1 *Regular Grid Map*. Grid map yang digunakan pada umumnya adalah grid map berbentuk 2D atau dapat juga disebut dengan regular grids. Salah satu contoh regular grid yang sering digunakan dalam berbagai penelitian perencanaan jalan adalah square grid dengan jaring yang berbentuk layaknya sebuah kotak. Contoh lainnya adalah hexagonal grid (berbentuk segi enam) dan triangular grid (berbentuk segi tiga). Penggunaan grid map selain square grid jarang ditemui di game-game pada umumnya.

Hanya beberapa game dengan fitur yang spesial seperti game dengan genre strategi yang menggunakan grid semacam ini [7].

3.2.1.2 *Irregular Grids*. Berbeda dengan grid regular yang berbentuk 2D, irregular grids menggunakan permukaan dari latar yang ada sebagai grid. Contoh dari irregular grids adalah visibility graphs, mesh navigation dan waypoints [7].

Grid Data	Map	Algorithms	Path Length	Time(s) ms
Obstacle Density 20%		A*	67	5.04
		Theta*	60.8	10.99
		C-Theta*	61.7	7.78
Obstacle Density 50%		A*	56.8	7.16
		Theta*	54.2	15.08
		C-Theta*	55.3	10.09
100 × 100				
Obstacle Density 20%		A*	117.5	12.12
		Theta*	110.8	19.02
		C-Theta*	111.5	16.13
Obstacle Density 50%		A*	107.5	7.57
		Theta*	106.1	11.6
		C-Theta*	106.1	10.6
100 × 100				
Maze		A*	172.8	22.79
		Theta*	158	53.37
		C-Theta*	163.5	31.95

Gambar 4. Tabel hasil penelitian C-Theta* [4]

Kemampuan C-Theta* dalam perencanaan jalannya dapat digunakan pada berbagai macam jenis grid dengan merubah sedikit dari jalan penelusuran algoritma C-Theta*. Hal ini dikarenakan C-Theta* merupakan algoritma yang memang dikhususkan untuk grid path planning dan lingkungan statis. Alasan lainnya adalah baik regular maupun irregular grids sama-sama menggunakan A* sehingga tidak menutup kemungkinan bagi algoritma C-Theta* untuk dapat diimplementasikan pada regular maupun irregular grids.

3.2.2. *Grid-based Games*. Salah satu contoh umum dari game berbasis grids adalah RTS (Real-Time Strategy). Game RTS memerlukan proses penghitungan perjalanan yang cepat antara 2 lokasi dari suatu peta. Beberapa game RTS juga memiliki unsur dinamis yang belum diuji-coba untuk C-Theta*. Setiap unit dalam

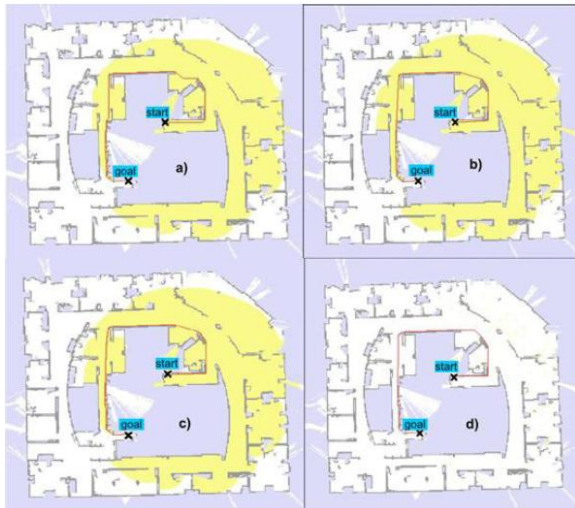
medan perang juga perlu mencari jalan tercepat menuju ke tempat persembunyian terdekat guna menghindari dari serangan musuh secara strategis [8].

3.3 Penggunaan C-Theta* pada Lingkungan dengan Kondisi Dinamis

3.3.1. *Mobile Robot*. Robot memerlukan algoritma pencarian jalan untuk bergerak melewati setiap objek yang ada pada dunia nyata. Dalam robot perencanaan jalan dikelompokkan menjadi 3 grup utama. *Motion planning, trajectory planning, dan planning under uncertainty*. Pengelompokkan ini memerlukan algoritma pencarian jalan yang digunakan dapat memenuhi ketiga hal diatas dengan bantuan aplikasi lainnya. Tanpa aplikasi seperti SLAM (*Simultaneous Localization and Mapping*) maka algoritma yang digunakan tidak dapat bekerja dengan baik. Hal ini dikarenakan adanya pergerakan sebuah robot ada pada lingkungan dengan kondisi yang dinamis, sehingga grids yang ada bukanlah sebuah pre-processed map (map yang sudah diproses untuk pengujian). Lingkungan dinamis ini menjadi masalah bagi algoritma *directed* seperti A* karena A* harus digunakan di map yang sudah di proses [9].

Pada tahun 2016, [10] membuat suatu penelitian mengenai perencanaan jalan robot dengan menggunakan algoritma A*. Hasil dari penelitian ini dapat terbilang belum cukup baik karena membutuhkan waktu lebih dari 2 detik untuk memproses perencanaan jalan. Masalah ini sebelumnya juga sudah dibuktikan oleh Duchon, et al. pada tahun 2014 bahwa algoritma A* sendiri memerlukan waktu yang lama untuk memproses perencanaan jalan bagi robot.

meneliti perencanaan jalan robot dengan menggunakan algoritma A* yang sudah dimodifikasi dengan algoritma Basic Theta* dan Phi* serta implementasi dari RSR (Rectangular Symmetric Reduction) dan JPS (Jump Point Search). Metode JPS ini melakukan pemotongan (crop) sel-sel yang ada di sekitar sel yang dievaluasi. Adanya metode ini mempercepat proses pencarian dari algoritma A*. Hasil dari penelitian [11] terbukti lebih baik dengan algoritma JPS (A*). (Gambar 5)



Gambar 5. Skenario pertama uji-coba algoritma a) A*; b) Basic Theta*; c) Phi*; d) JPS (A*) [11]

Modifikasi algoritma dengan menggunakan JPS memungkinkan algoritma C-Theta* untuk digunakan pada perencanaan jalan dari mobile robot ini, dikarenakan C-Theta* juga menggunakan A* sebagai bantuan pencariannya. Adanya Theta* yang sudah diuji oleh [11] juga menambah kemungkinan bagi *mobile robot* untuk bergerak menggunakan algoritma berbasis grids ini.

3.3.2. Aerial Robot. Aerial robot atau robot udara adalah robot yang dapat terbang seperti drone dengan kendali otomatis. Sistem navigasi pada robot udara adalah model 3D, dikarenakan variable ketinggian pada robot udara berpengaruh terhadap perencanaan jalan yang akan digunakan. Dalam hal ini C-Theta* tidak mungkin

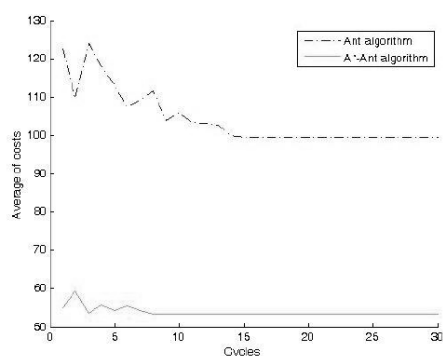
diimplementasikan pada robot udara. Robot udara memerlukan motion planning untuk pergerakan robot. Setiap putaran yang dilakukan oleh robot untuk berganti arah memerlukan waktu dan lengkungan yang tepat dalam prosesnya [12].

Peneliti lainnya yaitu [13] melakukan penelitian mengenai perencanaan jalan yang terbukti kuat dan aman bagi drone. Peneliti menggunakan metode SPP (Sequential Path Planning) dalam pengujian untuk kendaraan berjumlah banyak (pada kasus ini kendaraan berupa drone) dengan kondisi penerbangan yang berbeda-beda. Algoritma SPP ini dapat mengalokasikan lintasan ruang dan waktu bagi setiap kendaraan.

C-Theta* merupakan algoritma yang dikhususkan bagi *visibility graphs* dan grids, sedangkan algoritma yang digunakan oleh [12] dan [13] pada tahap pengujian adalah algoritma RTT dan SPP sebagai pengembangan dari RTT. Kedua algoritma tersebut dapat digunakan pada masalah perjalanan apapun [14]. Saat ini peneliti tidak dapat menemukan kemungkinan bagi algoritma C-Theta* untuk dapat diimplementasikan pada robot udara, namun untuk kedepannya ada kemungkinan bahwa C-Theta* dapat diimplementasikan pada robot udara dengan bantuan dari aplikasi dan algoritma pendukung lainnya.

3.3.3. Navigation System for Vehicle. Sistem navigasi saat berkendara merupakan hal yang penting bagi pengemudi kendaraan yang asing dengan lokasi dimana ia berada. Di era yang sudah modern ini, sistem navigasi diterapkan pada *electronic map* atau peta elektronik yang umumnya terdapat pada gadget khusus untuk sistem navigasi kendaraan.

Sebagian besar handphone pada era modern ini juga sudah memiliki peta elektronik *built-in* maupun peta elektronik lainnya yang dapat diunduh di application store. Perencanaan jalan berperan penting pada sistem navigasi kendaraan. Perencanaan jalan ini dapat berupa pencarian berbasis 2D maupun 3D yang mencari keadaan serta ruang (*state-space*) untuk mendeskripsikan sebuah lokasi pada peta. Beberapa contoh metode dan algoritma yang pernah diusulkan dalam pengembangannya adalah algoritma genetika, fuzzy, ant colonies, swarm, artificial bee colonies [15].



Gambar 6. Eksperimen terakhir rata-rata *costs* yang diperlukan Ant dan A*-Ant [15].

Pada tahun 2015, [16] meneliti sistem navigasi kendaraan dengan menggunakan algoritma A* yang digabungkan dengan algoritma ants colony (A*-ants). Algoritma A*-ants yang pernah diusulkan oleh peneliti adalah algoritma baru dimana algoritma A* dijalankan sebelum algoritma ants dan memperkuat feromon (zat kimia yang dikeluarkan sebagai jejak dari jalan yang pernah ditelusuri) dari hasil pencarian jalan yang dihasilkan algoritma ants. Jalur dengan feromon yang paling kuat ini yang nantinya akan diambil sebagai jalan tercepat. Hasil penelitian membuktikan bahwa A*-Ants bekerja lebih baik dibandingkan Ants sendiri (Gambar 6). Penelitian yang dilakukan oleh peneliti telah ditetapkan sebagai algoritma pada sistem navigasi dari

beberapa dari Kota Kerman di Iran. C-Theta* dapat diterapkan pada sistem navigasi yang serupa. Penggabungan C-Theta* dan ants memiliki kemungkinan untuk memberikan hasil yang lebih baik daripada penelitian sebelumnya dikarenakan C-Theta* juga menggunakan A* dan Theta*. Pertimbangan untuk menggunakan A* atau Theta* dalam memilih jalan yang akan dilalui dapat menimbulkan kemungkinan untuk menemukan jalan yang lebih baik.

3.4 Hasil Review

Hasil dari penelitian ini membuktikan bahwa kondisi lingkungan yang statis yaitu dengan menggunakan percobaan Aplikasi berbasis Grids (Regular, Irregular, Game-Based) dan Mobile Robot dapat memakai Algoritma C-Theta*. Hal ini dikarenakan jalan yang akan dilalui oleh pengguna sudah terbentuk dengan baik. Sedangkan pada percobaan Aerial Robot tidak dapat menggunakan algoritma C-Theta* sebagai algoritma path-planning-nya. Hal ini dikarenakan robot udara tidak mempunyai jalan yang pasti untuk mencapai suatu tujuannya. Selain itu koordinat Z atau altitude sangat berpengaruh terhadap jalannya proses ini.

4. Kesimpulan dan Pengembangan di Masa Depan

C-Theta* sebagai algoritma perencanaan jalan berbasis grids dapat diimplementasikan pada berbagai macam bidang dikarenakan adanya penggunaan A* dan Theta* yang sudah umum digunakan pada berbagai macam penelitian. C-Theta* juga dapat dibantu oleh beberapa algoritma dan aplikasi lainnya dalam implementasinya di lingkungan yang bersifat dinamis seperti pada game RTS tertentu dan robot.

Pengembangan dari C-Theta* dapat dilakukan dengan cara mengganti algoritma Theta* dengan variasi Theta* yang lebih baik seperti Lazy-Theta* yang pernah ditemukan oleh [5] atau variasi terbarunya yaitu Strict Theta* yang ditemukan oleh [17]. Dalam grid map dengan kompleksitas yang tinggi variasi dari Theta* ini juga dapat menerapkan penggabungan teori pemberatan dinamis dan metode hierarchical yang sudah dibuktikan oleh [18] pada tahun 2016. Penggabungan algoritma C-Theta* dengan hasil penelitian itu dapat meningkatkan performa dari C-Theta*.

DAFTAR PUSTAKA

- [1] Botea, A., Bouzy, B., Buro, M., Bauckhage, C., & Nau, D. (2014). Pathfinding in Games. *Artificial and Computational Intelligence in Games*, 6, 21–31
- [2] Barnouti, N., & Al-Dabbagh, S. (2016). Pathfinding in Strategy Games and Maze Solving Using a Search Algorithm. *Journal of Computer and*, 15–25.
- [3] Sinthamrongruk, T., Mahakitpaisarn, K., & Manopiniwes, M. (2013). A Performance Comparison between A* Pathfinding and Waypoint Navigator Algorithm on Android and iOS Operating System. *International Journal of Engineering and Technology*, 5(4), 498–501.
- [4] Mendonca, P., & Goodwin, S. (2015). C-Theta*: Cluster Based Path-Planning on Grids. 2015 International Conference on Computational Science and Computational Intelligence (CSCI), 605–608.
- [5] Nash, A., Koenig, S., & Tovey, C. (2010). Lazy Theta *: Any-Angle Path Planning and Path Length Analysis in 3D. *AAAI Conference on Artificial Intelligence*, 147–154.
- [6] Griffiths, M., Rogers, A., & Anderson, B. (2013). Migration, time and temporalities: review and prospect. *COMPAS Research Resources Paper*, (March), 1–31.
- [7] Abd Algfoor, Z., Sunar, M. S., & Kolivand, H. (2015). A comprehensive study on pathfinding techniques for robotics and video games. *International Journal of Computer Games Technology*, 2015.
- [8] Uras, T., Koenig, S., & Hernández, C. (2013). Subgoal Graphs for Optimal Pathfinding in Eight-Neighbor Grids. *Twenty-Third International Conference on Botea*, 224–232.
- [9] Lara-Cabrera, R., Cotta, C., & Fernandez-Leiva, A. J. (2013). A review of computational intelligence in RTS games. *Proceedings of the 2013 IEEE Symposium on Foundations of Computational Intelligence, FOCI 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, 114–121.
- [10] Graham, R., McCabe, H., & Sheridan, S. (2003). Pathfinding in Computer Games. *ITB Journal*, 4(8), 57–81.
- [11] Guruji, A. K., Agarwal, H., & Parsediya, D. K. (2016). Time-efficient A* Algorithm for Robot Path Planning. *Procedia Technology*, 23, 144–149.
- [12] Duchon, F., Babinec, A., Kajan, M., Beno, P., Florek, M., Fico, T., & Juri'ica, L. (2014). Path planning with modified A star algorithm for a mobile robot. *Procedia Engineering*, 96, 59–69.

- [12] Paranjape, A. A., Meier, K. C., Shi, X., Chung, S. J., & Hutchinson, S. (2013). Motion primitives and 3-D path planning for fast flight through a forest. *IEEE International Conference on Intelligent Robots and Systems*, 2940–2947.
- [13] Chen, M., Bansal, S., Tanabe, K., & Tomlin, C. J. (2017). Provably Safe and Robust Drone Routing via Sequential Path Planning: A Case Study in San Francisco and the Bay Area, 1–29.
- [14] Souissi, O., Benatitallah, R., Duvivier, D., Artiba, A., Belanger, N., & Feyzeau, P. (2013). Path Planning : A 2013 Survey. *Int. Conf. on Indust. Eng. and Sys. Manag.*, 2013(October), 1–8.
- [15] Singhal, P., & Kundra, H. (2014). A Review paper of Navigation and Pathfinding using Mobile Cellular Automata, 2(I), 43–50.
- [16] Salehinejad, H., & Nezamabadi-pour, H. (2015). Combined A*-ants algorithm: a new multi-parameter vehicle navigation scheme. 154–159.
- [17] Oh, S., & Leong, H. W. (2016). Strict Theta * Shorter Motion Planning Using Taut Paths, (Icaps), 253–257.
- [18] Le, P. T. H., & Lee, K. (2016). Weight Value and Map Complexity in Theta * Specific Weight - Dynamic Weight Theta *, 5003, 10–14.