

Algoritma Perencanaan Jalur Kendaraan Otonom di Lingkungan Perkotaan dari Sudut Pandang Filosofi Kuhn dan Filosofi Popper

Path Planning Algorithm for Autonomous Urban Vehicles from the Viewpoint of Kuhn's Philosophy and Popper's Philosophy

Muhammad Aria Rajasa Pohan^{1,*}, Bambang Riyanto Trilaksono¹, Sigit Puji Santosa²,
Arief Syaichu Rohman¹

¹Sekolah Teknik Elektro dan Informatika

²Sekolah Teknik Mesin dan Dirgantara

Institut Teknologi Bandung Jl. Ganesha No 10, Bandung

*Email : muhammad.aria@students.itb.ac.id

Abstrak – Kendaraan otonom dapat bermanfaat untuk meningkatkan keselamatan, efisiensi, aksesibilitas dan kenyamanan transportasi. Salah satu tugas penting yang harus dilakukan oleh kendaraan otonom adalah melakukan perencanaan jalur melalui lingkungan perkotaan yang dinamis dimana terdapat kendaraan lain dan pejalan kaki. Menurut filosofi Kuhn, sebuah keilmuan memiliki siklus hidupnya sendiri-sendiri. Siklus hidup suatu keilmuan terdiri dari fase kelahiran, fase pembuktian dan fase diterimanya keilmuan baru tersebut menjadi apa yang disebut sebagai ilmu normal. Demikian juga dengan perkembangan keilmuan perencanaan jalur, algoritma perencanaan jalur untuk kendaraan otonom masih terus diteliti dan dikembangkan oleh berbagai peneliti. Studi baru terus dilakukan dalam upaya menemukan teknik perencanaan jalur yang dapat diterima secara umum, sehingga algoritma tersebut akan menjadi ilmu normal. Menurut filosofi Popper, setiap algoritma yang diusulkan harus dapat diuji menggunakan prinsip falsifikasi untuk menentukan apakah teknik yang diusulkan tersebut akhirnya bisa menjadi ilmu normal atau tidak. Maka makalah ini bertujuan untuk memberikan gambaran mengenai siklus keilmuan perencanaan jalur kendaraan otonom. Makalah ini juga bertujuan untuk membandingkan penelitian-penelitian terkini mengenai algoritma perencanaan jalur untuk kendaraan otonom di daerah perkotaan. Dalam membandingkan algoritma-algoritma tersebut akan digunakan prinsip falsifikasi Popper. Dari hasil pengujian, ditunjukkan bahwa algoritma perencanaan jalur dapat bertahan dari metode falsifikasi Popper. Hasil falsifikasi terhadap algoritma perencanaan jalur ini menunjukkan penguatan nilai saintifik dari algoritma tersebut.

Kata Kunci : Kendaraan otonom, algoritma perencanaan jalur, lingkungan perkotaan, siklus keilmuan Kuhn, falsifikasi Popper

Abstract - Autonomous vehicles can be useful to improve safety, efficiency, accessibility and convenience of transportation. One important task that must be carried out by autonomous vehicles is to do path planning through a dynamic urban environment where there are other vehicles and pedestrians. According to Kuhn's philosophy, a science has its own life cycle. The science life cycle consists of the birth phase, the proof phase and the acceptance phase of the new science into what is called normal science. Likewise with the scientific development of path planning algorithms, path planning algorithms for autonomous vehicles are still being researched and developed by various researchers. New studies continue to be conducted in an effort to find pathway planning techniques that can be generally accepted, so that the algorithm will become a normal science. According to Popper's philosophy, each proposed algorithm must be tested using the principle of falsification to determine whether the proposed technique can eventually become a normal science or not. So this paper aims to provide an overview of the scientific cycle of planning autonomous vehicle lanes. This paper also aims to compare the latest research on path planning algorithms for autonomous vehicles in urban areas. In comparing these algorithms, the principle of Popper's falsification will be used. From the test results, it was shown that the path planning algorithm can survive the Popper falsification method. The results of the falsification of the path planning algorithm indicate the strengthening of the scientific value of the algorithm.

Keywords : Autonomous vehicles, path planning algorithms, urban environment, Kuhn's scientific cycles, Popper's falsification

I. PENDAHULUAN

Kendaraan otonom (*autonomous vehicle* atau *self-driving car*) adalah kendaraan yang mampu mengoperasikan dirinya sendiri, tanpa campur tangan manusia secara manual, untuk bergerak menuju tujuan yang telah ditentukan [1-3]. Agar kendaraan otonom dapat beroperasi tanpa campur tangan manusia, kendaraan otonom harus mampu mendeteksi kondisi lingkungan, memprediksi pergerakan objek di sekitarnya, dan membuat jalur ke tujuannya sambil menghindari rintangan statis maupun dinamis yang ada di sekitarnya [4]. Kendaraan otonom dapat bermanfaat untuk meningkatkan keselamatan, mengurangi kemacetan, meningkatkan efisiensi, meminimalkan ruang parkir, meningkatkan kenyamanan transportasi dan banyak manfaat lainnya [5-6].

Dengan adanya kemajuan dalam teknologi sensor dan komputasi, jumlah studi di bidang kendaraan otonom telah meningkat dalam tiga dekade terakhir [7]. Penelitian survei mengenai kendaraan otonom juga telah banyak dilakukan oleh para peneliti. Badue dkk telah melakukan penelitian survei mengenai mobil otonom yang dipublikasikan semenjak kompetisi DARPA *Urban Challenge* (DUC) serta memiliki sistem otomasi yang dapat dikategorikan oleh *Society of Automotive Engineers* (SAE) sebagai level 3 atau lebih. Makalah tersebut membahas arsitektur umum pada kendaraan otonom, sistem persepsi, sistem pengambilan keputusan, serta arsitektur sistem kendaraan otonom dari Universidade Federal do Espírito Santo (UFES) [8]. Zhao dkk. membahas empat teknologi utama yang digunakan pada kendaraan otonom. Teknologi-teknologi tersebut adalah sistem persepsi lingkungan, sistem perencanaan jalur, sistem navigasi kendaraan dan sistem kendali kendaraan. Paper ini juga membahas prediksi arah perkembangan penelitian dari kendaraan otonom [9]. Adapun Bimbrow membahas tren teknologi kendaraan otonom dari masa lalu, saat ini dan prediksi di masa depan. Dikarenakan saat ini telah banyak fitur semi-otonom yang diperkenalkan pada kendaraan modern (seperti *automatic braking*, *lane keeping*, dan *adaptive cruise control*), maka Bimbrow memprediksi bahwa dalam satu dekade lagi, sebagian besar perusahaan akan meluncurkan kendaraan otonom penuh [10]. Paden melakukan penelitian survei terkini mengenai algoritma perencanaan jalur dan kendali dari mobil otonom untuk lingkungan perkotaan. Paden mendiskusikan kelebihan dan kekurangan dari setiap algoritma perencanaan jalur [11].

Marina dan Sandu melakukan penelitian survei mengenai pemanfaatan *Deep Reinforcement Learning* (DRL) untuk kendaraan otonom. Paper tersebut membahas kelebihan dan kekurangan pemanfaatan DRL. Marina dan Sandu juga menyajikan penelitian tentang simulasi algoritma DRL pada kendaraan otonom [12]. Sementara Katrakazas dkk. melakukan perbandingan mengenai metode perencanaan jalur pada kendaraan otonom. Proses perencanaan jalur terdiri dari tiga tahapan, yaitu membangun jalur, menemukan manuver yang paling aman, serta menentukan jalur yang paling layak. Metode yang dikembangkan untuk masing-masing tahapan tersebut menunjukkan kompleksitas dan akurasi yang berbeda-beda. Katrakazas melakukan evaluasi kritis mengenai masing-masing metode ditinjau dari sisi kelebihan dan kekurangannya [13]. Meskipun banyak penelitian yang telah melakukan survei mengenai kendaraan otonom, sejauh pengetahuan penulis, belum ada penelitian yang membandingkan algoritma perencanaan jalur kendaraan otonom ini ditinjau dari sudut pandang filosofi Thomas S. Kuhn dan filosofi Karl Popper.

Tujuan dari penelitian ini adalah untuk melakukan survei literatur tentang algoritma perencanaan jalur kendaraan otonom di lingkungan perkotaan dari sudut pandang filosofi Thomas S. Kuhn dan dari sudut pandang filosofi Karl Popper. Literatur yang dipilih adalah literatur yang mengembangkan kendaraan otonom yang memenuhi kategori minimal level 3 menurut SAE. Selain menggunakan sumber-sumber dari publikasi ilmiah, penelitian ini juga melakukan penelusuran melalui situs web dari perusahaan yang mengembangkan kendaraan otonom serta melalui informasi dari berbagai media. Hal ini dilakukan dikarenakan tidak semua perusahaan komersial mempublikasikan hasil penelitian mereka melalui publikasi ilmiah. Artikel ini akan menyajikan siklus keilmuan dari algoritma perencanaan jalur berdasarkan filosofi Thomas S. Kuhn, serta menyajikan pula kajian falsifikasi berdasarkan filosofi Karl Popper terhadap berbagai algoritma perencanaan jalur.

II. METODE PENELITIAN

Penelitian ini menggunakan metode survei literatur. Literatur yang digunakan adalah literatur yang mempublikasikan penelitian mengenai sistem kendaraan otonom yang dapat dikategorikan oleh SAE sebagai level 3 atau lebih. Tetapi tidak semua perusahaan komersial yang mengembangkan kendaraan otonom

menginformasikan kemajuan penelitian mereka dalam publikasi ilmiah. Maka kami juga menggunakan situs perusahaan atau berita dari berbagai media yang membahas mengenai penelitian tentang kendaraan otonom. Diskusi akan ditekankan mengenai siklus keilmuan dari algoritma perencanaan jalur kendaraan otonom berdasarkan pandangan filosofi Thomas S. Khun, serta mengenai ketahanan berbagai algoritma perencanaan jalur kendaraan otonom terhadap berbagai macam pengujian, atau yang disebut sebagai falsifikasi berdasarkan pandangan filosofi Karl Popper. Analisis yang dilakukan meliputi kualitas performansi algoritma perencanaan jalur dalam menyelesaikan berbagai macam pengujian yang dilakukan.

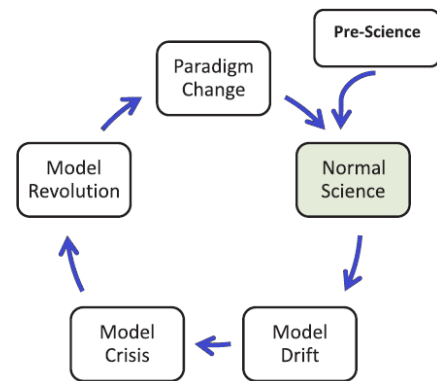
III. PERKEMBANGAN KEILMUAN PERENCANAAN JALUR BERDASARKAN SUDUT PANDANG FILOSOFI THOMAS S. KHUN.

Jika sebagian orang berpikir bahwa perkembangan sains adalah linear-akumulatif, maka Thomas S. Kuhn dalam bukunya yang berjudul "*The Structure of Scientific Revolution*," memiliki pandangan lain. Menurut Kuhn, sains memiliki siklus hidupnya sendiri. Siklus hidup sains terdiri dari fase kelahiran, fase pembuktian dan fase penerimaan dari sains baru ke dalam apa yang disebut sains normal. Siklus puncak suatu sains adalah dalam fase sains normal, yang kemudian usang karena digantikan oleh pengetahuan atau paradigma baru. Paradigma baru akan mengancam paradigma lama yang sebelumnya juga menjadi paradigma baru. Gambar 1 menunjukkan siklus hidup ilmu dalam filsafat Thomas S. Kuhn [14].

Pandangan Kuhn mengenai siklus ilmu dapat dilihat melalui tahapan-tahapan sebagai berikut.

9. *Pra-Science*

Pada tahap ini, kegiatan penelitian dilakukan secara terpisah dan tidak terorganisir. Ini dikarenakan tidak adanya paradigma tunggal yang diterima oleh semua ilmuwan tentang suatu teori. Ada beberapa ilmuwan yang mengusulkan paradigma baru. Peristiwa ini berlangsung selama periode waktu tertentu, sampai suatu paradigma tunggal diterima oleh semua ilmuwan. Ketika paradigma tunggal diterima, tahapan akan berlanjut menuju *Normal Science*.



Gambar 1. Siklus perkembangan keilmuan dari sudut pandang filosofis Thomas S. Khun (diadaptasi dari [14])

10. *Normal Science*

Pada tahap ini, aktivitas pengembangan ilmu menjadi terorganisir dan terarah. Pada tahap ini tidak ada perselisihan diantara para ilmuwan mengenai hal-hal mendasar dari paradigma tunggal tersebut. Paradigma tunggal dapat bertahan dari berbagai proses pengujian dan falsifikasi.

11. Anomali dan krisis

Tahap anomali terjadi ketika ada masalah yang tidak bisa diselesaikan oleh ilmu pengetahuan normal (paradigma tunggal yang sudah diterima tersebut). Jika semakin banyak anomali terjadi, dan berbagai kelompok ilmiah mempertanyakan kesempurnaan paradigma yang berlaku, maka sains telah memasuki tahap krisis sejak saat itu. Karena krisis, komunitas ilmiah akan mencoba mengembangkan paradigma baru untuk menyelesaikan krisis.

12. Revolusi sains, kembali ke ilmu normal, dan kembali terjadi krisis

Revolusi sains adalah tahap di mana paradigma lama digantikan oleh paradigma baru yang berbeda. Revolusi sains ini menunjukkan tahap perkembangan keilmuan yang non-kumulatif, tetapi terjadi secara revolusioner. Jika paradigma baru ini dapat diterima, maka paradigma baru ini akan menjadi ilmu normal baru, dan kemungkinan akan ditemukan anomali akan baru, krisis baru terjadi dan siklus keilmuan akan terus berulang.

Algoritma perencanaan jalur untuk kendaraan otonom masih merupakan ilmu yang berkembang. Oleh karena itu pada bagian ini akan dibahas perkembangan paradigma algoritma perencanaan jalur dari sudut pandang siklus perkembangan keilmuan Thomas S. Kuhn.

A. Perkembangan Keilmuan Algoritma Perencanaan Jalur Secara Umum

Algoritma perencanaan jalur telah melalui beberapa siklus paradigma. Salah satu paradigma pertama dalam algoritma perencanaan jalur adalah metode diskritisasi ruang kerja. Algoritma perencanaan jalur seperti Dijkstra [15] dan A* [16] merancang jalur optimal dalam ruang graph yang terkoneksi. Adapun algoritma D* [17] dan AD* [18] digunakan dalam graph yang dinamis. Tetapi penggunaan metode diskritisasi ini akan menyebabkan penurunan kinerja pada kasus berdimensi tinggi.

Metode-metode baru dikembangkan untuk menghindari proses diskritisasi ini. Salah satu paradigma yang diusulkan adalah metode berdasarkan *nature inspires artificial intelligence*. Metode-metode berbasis Logika Fuzzy [19-21], Jaringan Saraf Tiruan [22], Algoritma Genetik [23, 24], *Ant Colony Optimization* [25] dan Simulated Annealing [20] telah dikembangkan untuk aplikasi perencanaan jalur. Tetapi metode-metode ini dapat terjebak pada solusi yang bersifat minimal lokal, serta memiliki performansi yang buruk pada kasus lingkungan yang sempit.

Paradigma lainnya yang diusulkan adalah metode perencanaan reaktif berbasis sensor, seperti yang diusulkan oleh Borenstein dan Koren [26], Simmons [27] ataupun Belkhouche dan Bendjilali [28]. Tetapi perencanaan reaktif berbasis sensor ini tidak dapat dianggap sebagai perencanaan jalur global. Adapun metode kontrol klasik memerlukan formulasi model yang akurat dari robot maupun lingkungan [29, 30]. Formulasi model ini dapat menjadi pekerjaan yang rumit.

Paradigma perencanaan jalur lain yang diusulkan adalah perencanaan berbasis sampel atau *Sampling Based Planning* (SBP). SBP memiliki keunggulan dalam hal dapat memberikan solusi yang cepat pada masalah yang sulit [31]. Algoritma yang paling umum digunakan untuk SBP adalah *Probabilistic Roadmap Method* (PRM) [32] dan *Rapidly-exploring Random Tree* (RRT) [33]. Keunggulan algoritma PRM adalah dapat memberikan solusi yang bersifat asimptotik optimal, tetapi memiliki waktu komputasi yang tinggi, sedangkan keunggulan algoritma RRT adalah waktu komputasi yang rendah, tetapi memiliki hanya dapat memberikan solusi yang bersifat sub-optimal [34]. Algoritma RRT kemudian dikembangkan menjadi RRT* [35], yang memberikan solusi asimptotik optimal [36]. Namun, waktu komputasi RRT* masih tinggi. Saat ini ada banyak algoritma pengembangan dari RRT* ini. Akgun dan Stilman mengajukan

algoritma B-RRT* [36]. Metode ini meningkatkan optimasi jalur yang dihasilkan dengan cara menggunakan pencarian dua arah dari algoritma RRT*. Kumar et al. mengusulkan algoritma Goal-Bias-RRT [37]. Gammell et al. mengusulkan algoritma Informed RRT* [38], yang mengusulkan agar pengambilan sampel pada algoritma RRT hanya dilakukan pada daerah elips yang mengelilingi simpul awal dengan simpul akhir. Nasir et al. memperkenalkan algoritma RRT*-Smart untuk mempercepat laju konvergensi [39]. RRT*-smart menggunakan dua fitur yang dinamakan *intelligent sampling* dan *route optimization*.

Dari pembahasan diatas, dapat terlihat beberapa perubahan paradigma dalam algoritma perencanaan jalur. Beberapa paradigma tersebut adalah metode diskritisasi, metode berdasarkan *nature inspires artificial intelligence*, metode berdasarkan perencanaan reaktif berbasis sensor, metode kontrol klasik dan metode berbasis sampel.

B. Perkembangan Keilmuan Algoritma Perencanaan Jalur yang Digunakan pada Kendaraan Otonom

Gagasan mengenai kendaraan otonom dimulai pada tahun 1939 di New York World Fair dalam presentasi GM Futurama. Tetapi membutuhkan beberapa dekade untuk mewujudkan ide kendaraan otonom tersebut. Antara tahun 1987 hingga 1994, di Eropa telah dilaksanakan proyek EUREKA-PROMETHEUS dimana berbagai kendaraan dari perusahaan yang bermitra dengan proyek tersebut diotomatisasi. Sistem kontrol yang digunakan untuk mengikuti rute adalah metode Clothoid [40].

Pada 1997, program PATH bekerja sama dengan GM mendemonstrasikan delapan kendaraan otonom yang berjalan beriringan dengan jarak 6 meter antara satu dengan lainnya [41]. Carnegie Mellon University (CMU) juga mendemonstrasikan kendaraan otonom tunggal menggunakan kendaraan Navlab [42]. Pada tahun yang sama, Belanda mendemonstrasikan operasional CTS. CTS menyediakan layanan pengantaran *door-to-door* [43, 44]. Dalam sistem-sistem yang dibahas diatas, teknik yang digunakan adalah kontrol reaktif dan kerjasama kendaraan.

Salah satu penelitian pertama yang menguji teknik perencanaan jalur adalah proyek ARGO dari VisLab [45]. Sistem perencanaan jalur yang digunakan adalah pylonomial splines. Selanjutnya DARPA Challenge (Grand dan Urban Challenge) telah mendorong perkembangan teknik navigasi

otomatis dan teknik perencanaan jalur untuk kendaraan [46, 47].

Pemenang pertama dari DUC adalah tim Tartan Racing dari CMU. Tim Tartan Racing menggunakan berbagai variasi teknik untuk menghasilkan lintasan lokal di lingkungan terstruktur dan grafik kisi dalam ruang konfigurasi 4 dimensi (termasuk posisi, orientasi, dan kecepatan) bersama dengan algoritma Anytime D* untuk menemukan jalur bebas tabrakan di area parkir [48]. Pemenang kedua DUC adalah tim Stanford Racing. Tim Stanford menggunakan strategi perencanaan jalur Hybrid A* [49]. Pemenang ketiga dari DUC adalah tim VictorTango dari Virginia Tech. Tim VictorTango membangun diskritisasi grafik dari manuver yang memungkinkan dan mencari jalur menggunakan algoritma A* [50]. Tim yang berhasil menyelesaikan DUC di posisi keempat adalah tim dari Massachusetts Institute of Technology (MIT). MIT menggunakan varian dari algoritma RRT yang disebut Closed-Loop RRT dengan sampling bias [51]. Aspek inovatif utama dari sistem yang digunakan oleh tim MIT, dibandingkan dengan banyak tim lainnya, adalah bahwa keputusan otonom telah dibuat berdasarkan data persepsi lokal. Aspek inovatif lainnya adalah penggunaan algoritma perencanaan dan kontrol berbasis RRT yang mampu melakukan semua persyaratan mengemudi lajur, parkir dan bermanuver dengan pendekatan tunggal terpadu [52].

Kendaraan otonom pertama yang diproduksi Google adalah pada tahun 2009. Kendaraan otonom Google telah berhasil menempuh total jarak hingga 700.000 mil tanpa adanya kecelakaan yang disebabkan oleh sistem otomatisasi. Tetapi tidak ada publikasi ilmiah tentang teknik kontrol ataupun teknik perencanaan gerak yang digunakan oleh kendaraan otonom Google.

Pada tahun 2010, diselenggarakan VisLab Intercontinental Autonomous Challenge (VIAC) [53], yaitu tantangan kendaraan otonom dari VisLab untuk menempuh perjalanan sejauh 16.000 km dari Parma, Itali menuju Shanghai, Cina. Salah satu metode perencanaan jalur yang digunakan adalah metode clothoid [54]. Pada tahun 2010 dan 2012, Hyundai Motor Group menyelenggarakan kompetisi kendaraan otomatis di Korea. Tantangan pada tahun 2010 difokuskan pada penghindaran rintangan dan mengikuti jalur (*path following*). Pada tahun 2012, tantangan difokuskan pada pemahaman sistem otonom terhadap kondisi perkotaan dan mendeteksi sinyal lalu lintas [55]. Beberapa teknik perencanaan jalur

yang digunakan adalah clothoids [56, 57], dan RRT [58].

Salah satu kendaraan otonom yang telah disosialisasikan kepada masyarakat adalah mobil taksi otonom nuTonomy. Mobil nuTonomy beroperasi pada *Singapore's One-North technology business district*. Mobil nuTonomy ini menggunakan algoritma perencanaan jalur RRT* untuk mengevaluasi banyak alternatif jalur potensial berdasarkan data dari kamera maupun sensor lainnya [59]. Adapun Karaman dkk. telah menguji implementasi algoritma RRT* pada kendaraan otonom berupa robot forklift [60].

Melihat dari perkembangan implementasi algoritma perencanaan jalur yang digunakan pada kendaraan otonom diatas, dapat terlihat bahwa salah satu algoritma yang cukup sering digunakan adalah algoritma RRT. Maka selanjutnya akan ditinjau proses-proses pengujian yang pernah dilakukan pada algoritma RRT dan variasinya.

IV. ALGORITMA PERENCANAAN JALUR DARI SUDUT PANDANG FALSIFIKASI KARL POPPER

Popper berpendapat bahwa suatu teori atau pengetahuan baru dapat dipandang bersifat ilmiah jika dapat diuji (*testable*) melalui berbagai percobaan yang sistematis untuk menyangkalnya (*refutability*). Jika suatu teori dapat bertahan terhadap berbagai macam pengujian dan penyangkalan, maka kebenaran dari suatu teori tersebut semakin diperkokoh (Popper menyebutnya sebagai *corroboration*). Jika semakin besar upaya untuk menguji dan menyangkal suatu teori dan teori tersebut mampu untuk bertahan, maka akan semakin kokoh keberadaan teori tersebut [61].

Popper juga menegaskan bahwa setiap teori ilmiah hanya bersifat dugaan sementara (*conjecture*) dan tidak akan pernah tercapai kebenaran final. Setiap teori selalu memiliki kemungkinan untuk tergantikan oleh teori lain yang lebih tepat.

Berdasarkan pembahasan bab III, dapat terlihat bahwa salah satu algoritma perencanaan jalur yang sering digunakan pada kendaraan otonom adalah algoritma RRT. Maka pada bab IV ini akan dibahas pengujian-pengujian yang pernah dilakukan terhadap algoritma RRT dan variasinya ini. Kemampuan algoritma RRT untuk bertahan terhadap berbagai macam pengujian, menurut Karl Popper, akan menunjukkan kekokohan dari algoritma RRT ini.

Algoritma dasar dari RRT ditunjukkan pada **Gambar 2**. Adapun algoritma RRT* ditunjukkan pada **Gambar 3** – **Gambar 5**. Ilustrasi dari prinsip operasi *chooseparent* dan *rewire* pada RRT* ditunjukkan pada **Gambar 6**.

Algorithm 1 : $T = (V, E) \leftarrow RRT(q_{init})$

```

10.  $T \leftarrow InitializeTree()$ 
11.  $T \leftarrow InsertNode(\emptyset, q_{init}, T)$ 
12. for  $k \leftarrow 1$  to  $N$  do
13.    $q_{rand} \leftarrow RandomSample(k)$ 
14.    $q_{nearest} \leftarrow NearestNeighbor(q_{rand}, Q_{near}, T)$ 
15.    $q_{new} \leftarrow Steer(q_{nearest}, q_{rand}, \Delta q)$ 
16.   if  $Obstaclefree(q_{new}, q_{nearest})$  then
17.      $T \leftarrow InsertNode(q_{min}, q_{new}, T)$ 
18. end

```

Gambar 2. Algoritma dasar RRT

Algorithm 2 : $T = (V, E) \leftarrow RRT^*(q_{init})$

```

13.  $T \leftarrow InitializeTree()$ 
14.  $T \leftarrow InsertNode(\emptyset, q_{init}, T)$ 
15. for  $k \leftarrow 1$  to  $N$  do
16.    $q_{rand} \leftarrow RandomSample(k)$ 
17.    $q_{nearest} \leftarrow NearestNeighbor(q_{rand}, Q_{near}, T)$ 
18.    $q_{new} \leftarrow Steer(q_{nearest}, q_{rand}, \Delta q)$ 
19.   if  $Obstaclefree(q_{new}, q_{nearest})$  then
20.      $Q_{near} \leftarrow Near(T, z_{new})$ 
21.
22.    $q_{parent} \leftarrow ChooseParent(q_{new}, Q_{near}, q_{nearest})$ 
23.    $T \leftarrow InsertNode(q_{parent}, q_{new}, T)$ 
24.    $T \leftarrow Rewire(T, Q_{near}, q_{parent}, q_{new})$ 
25. end

```

Gambar 3. Algoritma RRT* (diadaptasi dari [62])

Algorithm 3 :

```

 $q_{min} \leftarrow ChooseParent(q_{rand}, Q_{near}, q_{nearest}, \Delta q)$ 
14.  $q_{min} \leftarrow q_{nearest}$ 
15.  $c_{min} \leftarrow Cost(q_{nearest}) + c(q_{rand})$ 
16. for  $q_{near} \in Q_{near}$  do
17.    $q_{path} \leftarrow Steer(q_{near}, q_{rand}, \Delta q)$ 
18.   if  $ObstacleFree(q_{path})$  then
19.      $c_{new} \leftarrow Cost(q_{near}) + c(q_{rand})$ 
20.     if  $c_{min} < c_{new}$  then
21.        $c_{min} \leftarrow c_{new}$ 
22.        $q_{min} \leftarrow q_{new}$ 
23.     end
24. end
25. end
26. return  $q_{min}$ 

```

Gambar 4. Operasi *chooseparent* pada algoritma RRT* (diadaptasi dari [62])

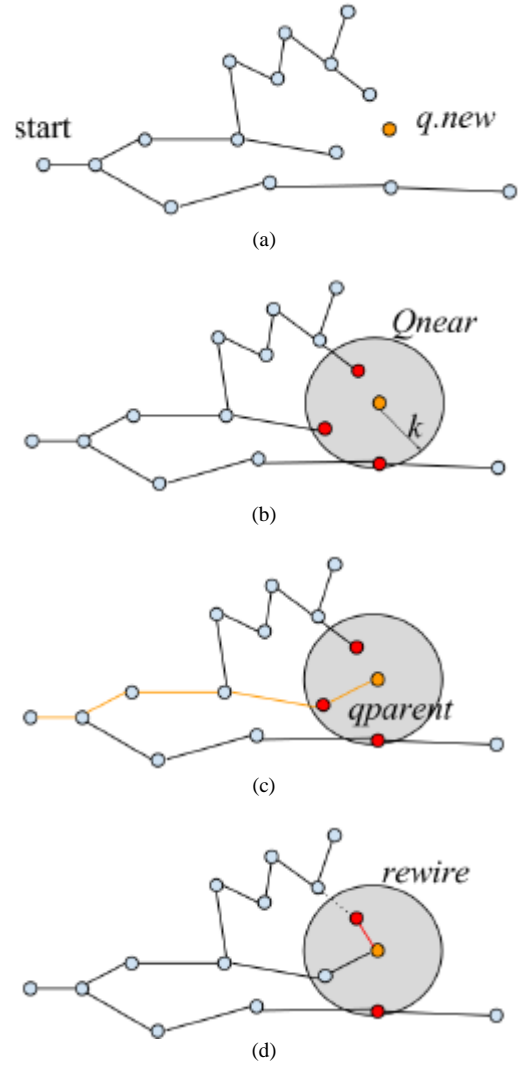
Algorithm 4 : $T \leftarrow Rewire(T, Q_{near}, q_{min}, q_{rand})$

```

7. for  $q_{near} \in Q_{near}$  do
8.    $q_{path} \leftarrow Steer(q_{near}, q_{rand}, \Delta q)$ 
9.   if  $ObstacleFree(q_{path})$  and
 $Cost(q_{rand}) + c(q_{path}) < Cost(q_{near})$  then
10.      $T \leftarrow ReConnect(q_{rand}, q_{near}, T)$ 
11. end
12. return  $T$ 

```

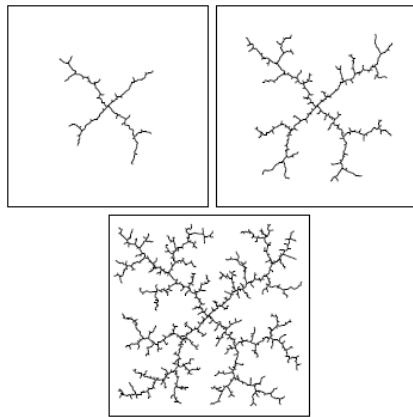
Gambar 5. Operasi *rewire* pada algoritma RRT* (diadaptasi dari [62])



Gambar 6. Ilustrasi dari prinsip kerja RRT* [31] (a) Node baru q_{new} ditentukan berdasarkan q_{rand} dan $q_{nearest}$. (b) Dipilih node-node terdekat dari q_{new} dan dinamakan Q_{near} . (c) Ditentukan q_{parent} , yaitu node dari Q_{near} yang akan memberikan jarak terdekat antara q_{new} dan node awal. (d) Node lainnya pada Q_{near} akan dilakukan operasi *rewire* terhadap q_{new} jika dapat dihasilkan jarak lebih dekat menuju node awal

A. Uji Completeness Algoritma RRT

Uji *completeness* bertujuan mengetahui apakah algoritma RRT dapat memberikan solusi jika solusi itu ada. LaValle pada publikasinya menyebutkan bahwa salah satu keunggulan dari algoritma RRT adalah memiliki sifat *probabilistically complete* [33]. Berdasarkan eksperimen, LaValle menunjukkan proses perkembangan pohon pencarian RRT seperti pada **Gambar 7**. Pohon pencarian RRT akan cenderung mengeksplorasi daerah yang belum terjelajahi. Maka jika waktu pencarian yang diberikan itu memungkinkan, maka algoritma RRT akan dapat selalu memberikan solusi atau algoritma RRT bersifat komplit secara probabilistic.



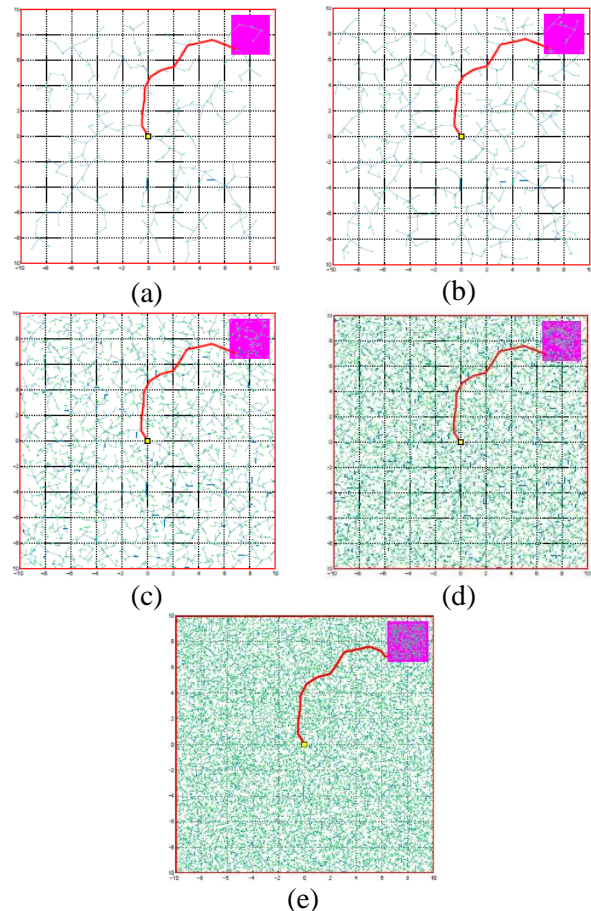
Gambar 7. Perkembangan pohon pencarian pada algoritma RRT akan cenderung mengeksplorasi daerah yang belum terjelajahi [33]

B. Uji *Optimality* Algoritma RRT dan RRT*

Uji *optimality* bertujuan mengetahui apakah algoritma RRT dan RRT* dapat memberikan solusi yang optimal. Karaman pada publikasinya menyebutkan bahwa algoritma RRT tidak bersifat optimal, sedangkan algoritma RRT* memiliki sifat *asymptotic optimal* [34]. Melalui eksperimen, Karaman menunjukkan perbandingan pengujian algoritma RRT dan RRT* pada area tanpa rintangan, seperti yang ditunjukkan pada **Gambar 8** dan **Gambar 9**. Pada **Gambar 8** menunjukkan bahwa meskipun jumlah iterasi terus ditambah, algoritma RRT tidak cenderung menghasilkan jalur yang optimal (jalur optimal pada area tanpa rintangan adalah garis lurus yang menghubungkan node awal dengan node akhir). Adapun pada **Gambar 9** menunjukkan bahwa dengan menambahkan jumlah iterasi, algoritma RRT* dapat cenderung menghasilkan jalur yang optimal (jalur yang dihasilkan algoritma RRT*, jika iterasi ditambah, akan cenderung menghasilkan garis lurus yang menghubungkan node awal dan node akhir).

C. Uji Konvergensi Algoritma RRT, RRT*, dan Informed RRT*

Uji konvergensi dari algoritma RRT dan RRT* telah dilakukan oleh LaValle [33], Karaman [34], dan Guevara [63]. Karaman melakukan pengujian dengan cara kedua algoritma dijalankan 20.000 iterasi sebanyak 500 kali. Lalu total jarak dari jalur terbaik dirata-ratakan pada tiap iterasi. Hasilnya ditunjukkan pada **Gambar 10**. Dapat terlihat bahwa algoritma RRT hanya dapat mendekati dalam skala faktor $\sqrt{2}$ dari solusi optimal (LaValle juga melaporkan hasil yang mirip). Adapun algoritma RRT* dapat konvergen menuju solusi optimal.

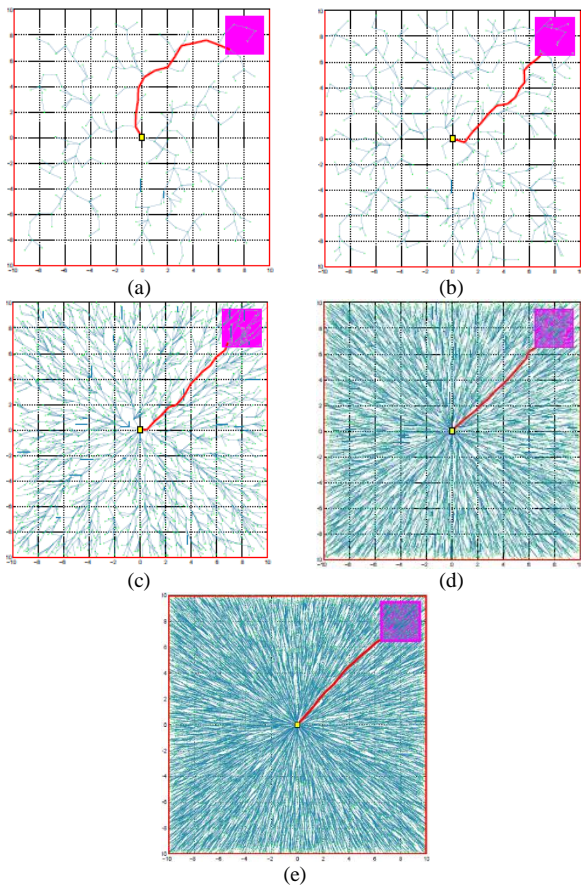


Gambar 8. Pengujian algoritma RRT pada area tanpa hambatan. (a) 250 cabang, (b) 500 cabang, (c) 2500 cabang, (d) 10.000 cabang dan (e) 20.000 cabang [34]

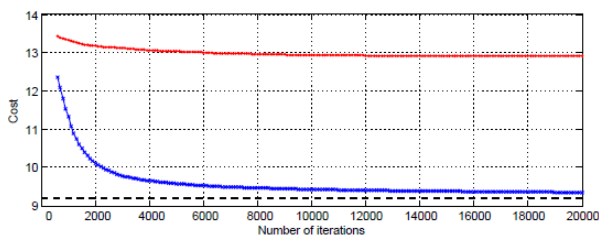
Mirip dengan yang dilaporkan oleh Karaman, Guevara juga melakukan uji konvergensi algoritma RRT, RRT* dan Informed RRT* pada kasus area yang memiliki jalur yang sempit (*narrow gap*). Hasilnya ditunjukkan pada **Gambar 11**. Terlihat bahwa algoritma RRT tidak menunjukkan peningkatan kualitas jalur, adapun algoritma RRT* dan Informed RRT* menunjukkan peningkatan kualitas jalur. Pada kasus tersebut, algoritma informed RRT* lebih dahulu berhasil melewati jalur yang sempit daripada algoritma RRT*.

D. Pengujian Algoritma RRT* dan Variasinya pada Lingkungan Khusus

Terdapat beberapa lingkungan khusus, yang telah diuji oleh beberapa peneliti, yang menunjukkan bahwa variasi dari RRT* memiliki keunggulan dibandingkan RRT* pada lingkungan tersebut. Beberapa lingkungan khusus yang menjadi *benchmark* pengujian algoritma perencanaan jalur adalah *maze* (labirin), *narrow* (celah sempit), *trap* (ruangan dengan celah sempit untuk masuk atau keluar) serta *clutter* (lingkungan dengan banyak rintangan).

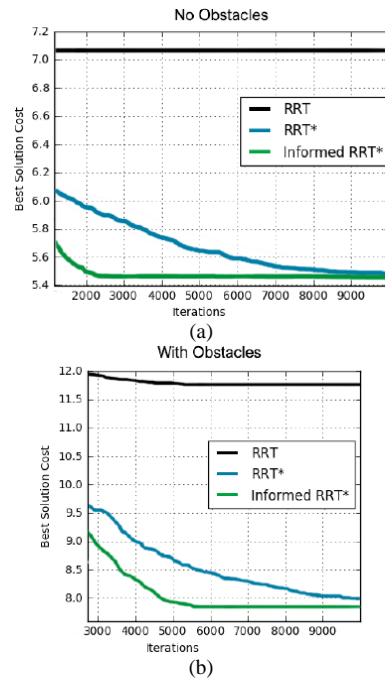


Gambar 9. Pengujian algoritma RRT* pada area tanpa hambatan. (a) 250 cabang, (b) 500 cabang, (c) 2500 cabang, (d) 10.000 cabang dan (e) 20.000 cabang [34]

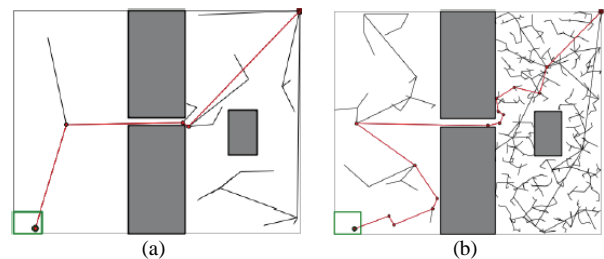


Gambar 10. Total jarak terbaik dari algoritma RRT (garis merah atas) dan algoritma RRT* (garis biru bawah). Jalur optimal ditunjukkan oleh garis hitam putus-putus [34]

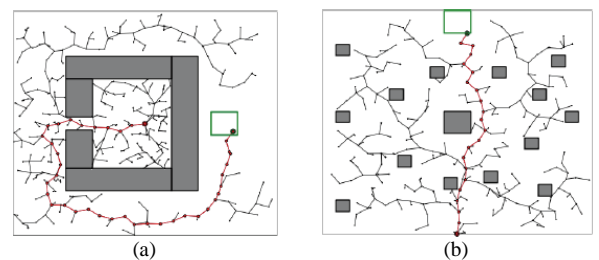
Elbanhawi [31] melaporkan bahwa algoritma dasar RRT tetap bisa menemukan jalur menuju titik tujuan pada kasus *narrow*, *trap*, maupun *clutter* seperti yang ditunjukkan pada **Gambar 12** dan **Gambar 13**. Memang ada faktor keberuntungan pada kasus-kasus tersebut. Seperti yang dicontohkan pada **Gambar 12**, jika beruntung, maka algoritma RRT dapat cukup cepat untuk melalui celah sempit dan mencapai titik tujuan seperti yang ditunjukkan pada **Gambar 12 (a)**. Tetapi adakalanya algoritma RRT membutuhkan waktu cukup lama untuk dapat melalui celah sempit seperti yang ditunjukkan pada **Gambar 12 (b)**. Adapun Nasir [39] melaporkan bahwa algoritma RRT* mampu membangun jalur pada kasus labirin (*maze*).



Gambar 11. Grafik total jarak terhadap iterasi dari algoritma RRT, RRT*, dan informed RRT*. Ketiga algoritma dijalankan 10.000 iterasi sebanyak 100 kali, (a) area tanpa rintangan, (b) area dengan rintangan [63]



Gambar 12. Algoritma RRT mampu membangun jalur yang melewati celah sempit (*narrow*). (a) jika beruntung maka algoritma RRT dapat cepat melewati celah sempit, (b) tetapi algoritma RRT terkadang membutuhkan waktu cukup lama untuk melewati celah sempit [31]

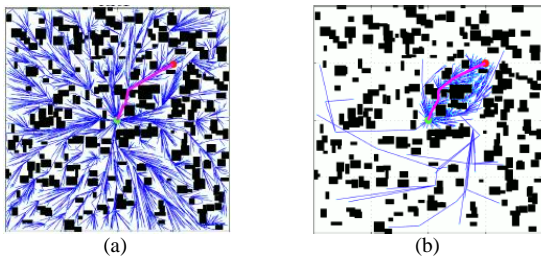


Gambar 13. (a) Algoritma RRT mampu membangun jalur pada kasus *trap* maupun (b) *clutter*[31]

Walaupun algoritma RRT dan RRT* mampu membangun jalur pada kasus *maze*, *labirin*, *narrow*, dan *clutter*, beberapa peneliti melaporkan performansi yang lebih baik untuk menyelesaikan kasus-kasus tersebut, jika menggunakan algoritma variasi dari RRT dan RRT* ini. Nasir [39] melaporkan bahwa penggunaan algoritma RRT*-Smart dapat menghasilkan jalur yang lebih pendek dibandingkan penggunaan algoritma

RRT* untuk kasus *maze*, *labirin*, *narrow*, dan *clutter*. Noreen [64] juga melaporkan informasi yang sama, yaitu penggunaan algoritma RRT-Smart dapat mengurangi waktu komputasi dari RRT* dan dapat menghasilkan jalur yang lebih pendek dari algoritma RRT*.

Gammel [38] melaporkan bahwa penggunaan algoritma Informed RRT* dapat mempercepat waktu komputasi dibandingkan penggunaan algoritma RRT*, seperti yang ditunjukkan pada **Gambar 14**. Klemm [65] juga melaporkan bahwa penggunaan algoritma RRT*-Connect dapat mempercepat waktu komputasi dibandingkan algoritma RRT*. Jika strategi yang diusulkan oleh algoritma informed-RRT* adalah membatasi daerah pengambilan sampel hanya pada daerah elips yang mengelilingi simpul awal dengan simpul akhir, maka strategi yang diusulkan oleh algoritma RRT*-Connect adalah menggunakan pencarian dua arah dari algoritma RRT*, yaitu pencarian dimulai secara bersamaan dari titik awal dan titik akhir.

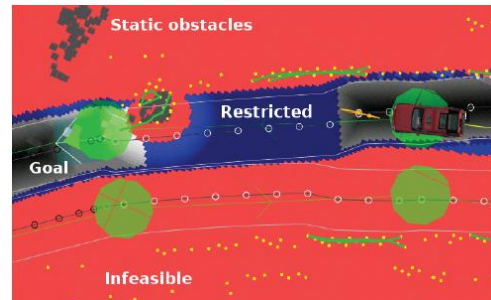


Gambar 14. (a) Algoritma RRT* dapat mencapai solusi optimal dalam waktu 8,26 detik, (b) algoritma Informed RRT* dapat mencapai solusi optimal dalam waktu 0,76 detik [38].

E. Pengujian Algoritma RRT* pada Perencanaan Jalur Kendaraan Otonom

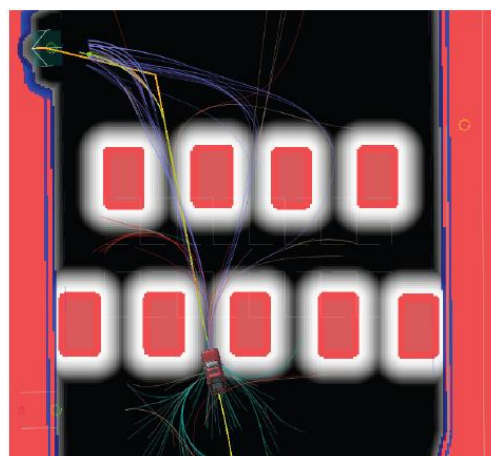
Kuwata [51] telah melaporkan berbagai pengujian variasi algoritma RRT* yaitu algoritma *Closed-Loop* RRT* untuk implemmentasi pada perencanaan jalur kendaraan otonom. Untuk mengevaluasi apakah lintasan yang direncanakan tidak akan menabrak rintangan atau melanggar aturan jalan raya, maka digunakan peta *drivability*. Semua data persepsi dari sensor, termasuk hambatan statis, hambatan dinamis, batas-batas jalur dan bahaya yang ada dipermukaan jalan disajikan dalam peta *drivability* ini. **Gambar 15** menunjukkan tampilan dari peta *drivability*. Wilayah berwarna merah mewakili wilayah yang tidak dapat dilalui (karena keluar jalur, adanya hambatan, dll.). Wilayah hitam adalah wilayah yang dapat dilalui dengan aman. Wilayah abu-abu, atau putih berarti wilayah tersebut masih dapat dilalui tetapi akan mendapat penalti. Wilayah putih berarti berpenalti tinggi. Nilai

penalti diskalakan dengan warna skala abu-abu. Adapun wilayah biru dinamakan "terbatas", yaitu kendaraan dapat melaju tetapi tidak boleh berhenti. Contoh penggunaan daerah terbatas ini adalah saat terdapat hambatan di jalan. Kendaraan akan berhenti dengan jarak yang cukup ke hambatan, seperti yang ditunjukkan pada **Gambar 15**. Saat jalur telah memungkinkan untuk dilewati, maka kendaraan akan melewati area terbatas tersebut dan melewati hambatan yang ada di jalur.



Gambar 15. Peta *drivability* dan penjelasannya [51]

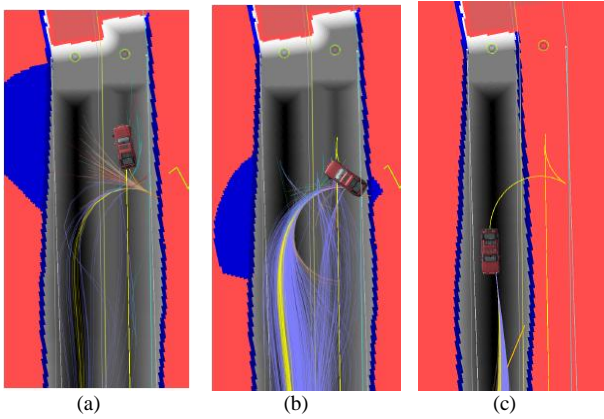
Gambar 16 menunjukkan proses navigasi wilayah yang memiliki hambatan. Garis oranye adalah masukan ke pengontrol. Masukan tersebut dihasilkan oleh algoritma pengambilan sampel. Garis hijau menunjukkan lintasan prediksi yang memungkinkan. Warna ungu yaitu mencapai target. Warna coklat muda yaitu aman tetapi tidak mencapai target. Warna merah yaitu tidak aman. Terlihat bahwa beberapa jalur telah berhasil mencapai tujuan. Dapat terlihat bahwa jalur yang dihasilkan oleh algoritma *Closed-Loop* RRT ini telah memberikan jalur yang halus.



Gambar 16. Perencanaan jalur di area dengan hambatan. Titik start adalah di bagian bawah, dan tujuannya ada di sudut kiri atas [51]

Gambar 17 menunjukkan pengujian untuk melakukan putaran balik (*U-turn*). Melakukan putaran balik ini merupakan manuver yang cukup rumit. Sistem membangun tiga proses jalur yang

saling berhubungan. Jalur pertama mengarahkan kendaraan untuk bergerak ke kiri-depan untuk memulai belokan. Jalur kedua mengarahkan kendaraan untuk bergerak mundur ke kanan. Jalur ketiga mengarahkan kendaraan untuk bergerak ke kiri depan untuk menyelesaikan perputaran.



Gambar 17. Perencanaan jalur untuk melakukan manuver putar balik. Gambar 17a, 17b, dan 17c menunjukkan tiga tahapan jalur yang dirancang untuk melakukan putaran balik [51]

V. KESIMPULAN

Makalah ini membahas penelitian survey mengenai algoritma perencanaan jalur dari sudut pandang filsafat ilmu Thomas S. Khun dan Karl Popper. Berdasarkan sudut pandang filsafat Thomas S. Khun, telah dibahas perkembangan paradigma perencanaan jalur secara umum maupun perkembangan paradigma perencanaan jalur yang digunakan pada kendaraan otonom. Adapun berdasarkan sudut pandang Karl Popper, telah dibahas berbagai pengujian atau usaha falsifikasi yang pernah dilakukan pada algoritma-algoritma perencanaan jalur yang ada. Pengujian yang dibahas pada paper ini meliputi uji *completeness*, uji *optimality* dan uji konvergensi. Algoritma perencanaan jalur dapat menunjukkan bahwa algoritma tersebut memiliki sifat *completeness*, *optimality* dan konvergen menuju nilai optimal. Hasil falsifikasi terhadap algoritma perencanaan jalur ini menunjukkan penguatan nilai saintifik dari algoritma tersebut.

DAFTAR PUSTAKA

- [1] Y.S. Dawood, M. Almageed, dan A. Mahmood, "Autonomous Model Vehicles: Signal Conditioning and Digital Control Design," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 8, no.3, pp. 18-24, 2018
- [2] C. Berger, "From a competition for self-driving miniature cars to a standardized experimental platform: concept, models, architecture, and evaluation," *Journal of Software Engineering for Robotics*, vol. 5, no. 1, pp. 63-79, 2014
- [3] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Weling, dan S. Thrun, "Towards fully autonomous driving: Systems and algorithms," *2011 IEEE Intelligent Vehicles Symposium*, vol. 32, no. 14, pp. 163-168, 2011
- [4] R. Lemos, O. Garcia, dan J. V. Ferreira, "Local and global path generation for autonomous vehicles using splines," *Ingenieria* vol. 21, no. 2, pp. 188-200, 2016
- [5] S. Trubia, T. Giuffre, A. Canale, dan A. Severino, "Automated Vehicle: a Review of Road Safety Implications as Driver of Change," *27th CARSP Conference*, pp. 1-15, 2015
- [6] J. Bierstedt, A. Gooze, C. Gray, J. Peterman, L. Raykin, dan J. Walters, "Effects of next-generation vehicles on travel demand and highway capacity," *FP Think Working Group*, pp. 10-11, 2014
- [7] B. Paden B, M. Čáp, S. Z. Yong, D. Yershov D, dan E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, no. 1, no. 1, pp. 33-55, 2016
- [8] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. F. R. Jesus, R. F. Berriel, T. M. Paixão, F. Mutz, T. Oliveira-Santos, dan A. F. D. Souza, "Self-Driving Cars: A Survey," *arXiv preprint arXiv:1901.04407*, pp. 1-31, 2019
- [9] J. Zhao, B. Liang, dan Q. Chen, "The key technology toward the self-driving car," *International Journal of Intelligent Unmanned Systems*, vol. 6, no. 1, pp. 2-20, 2018
- [10] K. Bimbrav K, "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 1, pp. 191-198, 2015
- [11] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, dan E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33-55, 2016
- [12] L. Marina, dan A. Sandu, "Deep Reinforcement Learning for Autonomous Vehicles-State of the Art," *Bulletin of the Transilvania University of Brasov Engineering Sciences Series I*, vol. 10, no. 2, pp. 195-202, 2017
- [13] C. Katrakazas, M. Qudus, W. H. Chen, dan L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416-442, 2015
- [14] T. S. Kuhn, "The Structure of Scientific Revolutions," *Chicago, IL: University of Chicago Press*, 1962
- [15] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269-271, 1959
- [16] P. E. Hart, N. J. Nilsson, dan B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968
- [17] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," *International Journal of Robotics and Automation*, vol. 10, pp. 89-100, 1995
- [18] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, dan S. Thrun, "Anytime search in dynamic graphs," *Artificial Intelligence*, no. 172, vol. 14, pp. 1613-1643, 2008.
- [19] H. R. Beom dan H. S. Cho, "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning," *IEEE Transactions on Systems Man and Cybernetics*, vol. 25, no. 3, pp. 464-477, 1995
- [20] H. Martínez-Alfaro dan S. Gómez-García, "Mobile robot path planning and tracking using simulated annealing and fuzzy logic control," *Expert Systems with Applications*, vol. 15, no. 3, pp. 421-429, 1998.
- [21] H. Seraji dan A. Howard, "Behavior-based robot navigation on challenging terrain: A fuzzy logic approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 308-321, 2002
- [22] D. Janglová, "Neural networks in mobile robot motion," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 15-22, 2004
- [23] M. Gerke, "Genetic path planning for mobile robots," *Proceedings of the 1999 American Control Conference*, vol. 4, pp. 2424-2429, 1999

- [24] Y. Hu, S. X. Yang, L. Z. Xu, dan M. H. Meng, "A Knowledge Based Genetic Algorithm for Path Planning in Unstructured Mobile Robot Environments," *IEEE International Conference on Robotics and Biomimetics*, pp. 767-772, 2004
- [25] M. A. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, dan P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing Journal*, vol. 9, no. 3, pp. 1102-1110, 2009
- [26] J. Borenstein dan Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*, vol. 7, no. 3, pp. 278-288, 1991
- [27] R. Simmons, "The curvature-velocity method for local obstacle avoidance," *Proceedings of IEEE international conference on robotics and automation*, vol. 4, pp. 3375-3382, 1996
- [28] F. Belkhouche dan B. Bendjilali, "Reactive Path Planning for 3-D Autonomous Vehicles," *IEEE transactions on control systems technology*, vol. 20, no. 1, pp. 249-256, 2012
- [29] T. M. Howard dan A. Kelly, "The International Journal of Robotics Research," vol. 26, no. 2, pp. 141-166, 2007
- [30] M. Werling, S. Kammel, J. Ziegler, dan L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346-359, 2012
- [31] M. Elbhanawi, dan M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56-77, 2014
- [32] L. E. Kavraki, P. Svestka, J. C. Latombe, dan M. H. Overmars, M. H. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566-580, 1996
- [33] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Tech. Rep., Computer Science Dept., Iowa State University*, pp. 1-4, 1998
- [34] S. Karaman, dan E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846-894, 2011
- [35] S. Karaman, dan E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," *49th IEEE conference on decision and control (CDC)*, pp. 7681-7687, 2010
- [36] B. Akgun, dan M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2640-2645, 2011
- [37] B. S. Kumar, P. S. Abhimanyu, B. V. V. P. Bharagav, P. Agarwal, dan K. M. Krishna, *RoboCup SSL Team Description, IRL RC*, pp. 1-4, 2010
- [38] J. D. Gammell, S. S. Srinivasa, dan T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2997-3004, 2014
- [39] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, dan M. S. Muhammad, "RRT*-SMART: A rapid convergence implementation of RRT," *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, pp. 1-12, 2013
- [40] R. Behringer dan N. Muller, "Autonomous road vehicle guidance from autobahnen to narrow curves," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 810-815, 1998
- [41] S. Shladover, "The gm-path platoon scenario," *Intellimotion*, vol. 6, no. 3, 1997
- [42] C. Thorpe, T. Jochem, dan D. Pomerleau, "Automated highways and the free agent demonstration," *Robotics Research. Springer*, pp. 246-254, 1998
- [43] K. M. Malone, J. W. Van Der Wiel, dan B. A. Saugy, "Cybernetic transport systems: lessons to be learned from user needs analysis and field experience," *2002 IEEE Intelligent Vehicle Symposium*, vol. 2, pp. 551-556, 2002
- [44] M. Parent, "Advanced urban transport: Automation is on the way," *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 9-11, 2007
- [45] A. Piazzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, dan A. Broggi, "Quintic G/sup 2/-splines for the iterative steering of vision-based au-tonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27 - 36, 2002
- [46] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, "Toward reliable off road autonomous vehicles operating in challenging environments," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449-483, 2006
- [47] E. Krotkov, S. Fish, L. Jackel, B. McBride, M. Perschbacher, dan J. Pippine, "The DARPA PerceptOR evaluation experiments," *Autonomous Robots*, vol. 22, no. 1, pp. 19-35, 2007
- [48] C. Urmsion, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolsi, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, dan M. Gittleman, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425-66, 2018
- [49] M. Montemerlo, J. Becker, S., Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, dan D. Haehnel, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569-597, 2008
- [50] C. Reinholdt, D. Hong, A. Wicks, A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern, dan M. Webster, "Odin: Team victortango's entry in the darpa urban challenge," *Journal of field Robotics*, vol. 25, no. 8, pp. 467-92, 2008
- [51] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, dan J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105-1118, 2009
- [52] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, dan A. Nathan, "The MIT-Cornell collision and why it happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775-807, 2008
- [53] M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, A. Coati, S. Debatisti, A. Falzoni, dan R.I. Fedriga, "VIAC: An out of ordinary experiment," *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 175-180, 2011
- [54] A. Broggi, P. Medici, P. Zani, A. Coati, dan M. Panciroli, "Autonomous vehicles control in the VisLab intercontinental autonomous challenge," *Annual Reviews in Control*, vol. 36, no. 1, pp. 161-171, 2012
- [55] K. Jo, J. Kim, D. Kim, C. Jang, dan M. Sunwoo, "Development of autonomous car - Part I: Distributed system architecture and development process," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, 2014
- [56] K. Jo, J. Kim, D. Kim, C. Jang, dan M. Sunwoo, "Development of autonomous car - Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, 2015
- [57] K. Chu, M. Lee, dan M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599-1616, 2012.
- [58] K. Jo, M. Lee, D. Kim, J. Kim, C. Jang, E. Kim, S. Kim, D. Lee, C. Kim, S. Kim dan K Huh, "Overall reviews of autonomous vehicle ai-system architecture and algorithms," *IFAC Proceedings*, vol. 46, no. 10, pp. 114-119, 2013
- [59] E. Ackerman, "Hail, robo-taxi!" *IEEE Spectrum*, vol. 54, no. 1, pp. 26-29, 2017
- [60] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, dan S. Teller, "Anytime motion planning using the RRT," *2011 IEEE International Conference on Robotics and Automation*, pp. 1478-1483, 2011
- [61] K. Popper, "The Logic of Scientific Discovery," *Routledge*, 2005
- [62] D. Connell, dan H. M. La, "Dynamic path planning and replanning for mobile robots using RRT," *2017 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1429-1434, 2007
- [63] B. C. Guevara, "An Overview of the Class of Rapidly-Exploring Random Trees," *Utrecht University*, 2018

[64] I. Noreen, A. Khan, dan Z. Habib, "A comparison of RRT, RRT* and RRT*-smart path planning algorithms," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 10, pp. 20-27, 2016

[65] S. Klemm, J. Oberländer, A. Hermann, A. Roennau, T. Schamm, J.M. Zollner, dan R. Dillmann, R., "RRT*-Connect: Faster, asymptotically optimal motion planning," *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1670-1677, 2015