

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Masalah perencanaan jalur adalah bidang penelitian mendasar dalam dunia robotika [1]. Masalah perencanaan jalur adalah untuk menemukan lintasan yang membawa robot dari keadaan awal (*start*) ke keadaan tujuan (*goal*) sambil menghindari tabrakan dengan rintangan [2]. Terdapat berbagai aplikasi dalam perencanaan jalur, Brian Paden [3] menggunakan perencanaan jalur pada *self driving vehicle*, perilaku mengemudi yang berupa *change lane*, *cruise in lane*, atau *turn right* harus diterjemahkan ke dalam jalur atau lintasan. Jalur lintasan yang dihasilkan layak secara dinamis, dan menghindari tabrakan dengan rintangan yang terdeteksi oleh sensor *on-board*. David Gonzales [4] dalam penelitiannya pada *automated vehicle* menggunakan teknik perencanaan yang diklasifikasikan dalam 4 kelompok, menurut penerapannya dalam penggerak otomatis: pencarian grafik, pengambilan sampel, interpolasi, dan optimisasi numerik. Ying Liu [5] melakukan penelitian algoritma perencanaan jangkauan *real-time* berbasis heuristik untuk sosok manusia virtual. Posisi awal dan tujuan dibuat dalam ruang kerja 3D, Menghitung jalur bebas tabrakan yang menentukan semua konfigurasi lengan manusia untuk bergerak dari awal ke tujuan.

Berbagai metodologi telah dilakukan untuk menghasilkan jalur yang optimal [6], perencana jalur berbasis pohon dikembangkan oleh LaValle dan Kuffner [7] yang berguna untuk menjelajahi ruang kerja atau C-space, teknik perencanaan jalur ini dikenal dengan *Rapidly Exploring Random Tree (RRT)*. *Rapidly Exploring*

*Random Tree* (RRT) diperkenalkan pada penelitian [8] sebagai algoritma perencanaan untuk dengan cepat mencari ruang berdimensi tinggi yang memiliki kendala aljabar (yang muncul dari hambatan) dan kendala diferensial (yang muncul dari nonholonomi dan dinamis). Pada penelitian [9] diterangkan bahwa *Rapidly Exploring Random Tree* (RRT) memiliki keunggulan dalam memberikan jalur pertama tercepat dan juga kelengkapan probabilistik tetapi tidak menjamin optimalitas asimtotik. Pada tahun 2010, Karaman dan Frazzoli pertama kali mengusulkan algoritma *Rapidly Exploring Random Tree\** (RRT\*) untuk mengatasi masalah probabilitas dari algoritma *Rapidly Exploring Random Tree* (RRT) yang belum optimal [10].

Salah satu bagian dari algoritma perencanaan jalur adalah metode *sampling*, terdapat beberapa metode *sampling* yang telah diusulkan untuk mengurangi waktu konvergensi yang telah diterapkan pada navigasi perencanaan jalur [11]. Penelitian Jing Dong menunjukkan bagaimana interpolasi *Gaussian Process* dapat lebih meningkatkan kecepatan algoritma, dan menggunakan grafik faktor serta optimasi numerik untuk melakukan intervensi dengan cepat [12]. Penelitian [13] menggunakan metode *Boundary Sampling* yang mengklasifikasikan hambatan pada ruang kerja menjadi hambatan yang berharga dan yang tidak berharga, kemudian menggunakan informasi batasnya untuk membuat titik sampel perencanaan jalur di sekitarnya, selanjutnya pohon acak akan tumbuh ke posisi tujuan. Terdapat beberapa *library path planning* pada bahasa pemrograman, contohnya penelitian [14] melakukan implementasi algoritma pada Borland C++, kemudian diuji menggunakan bahasa pemrograman *visual basic* dan *delphi*. Sejauh yang penulis ketahui, belum ada *library path planning* pada bahasa pemrograman Labview. Pada

penelitian Tugas Akhir ini akan dilakukan metode integrasi, yaitu metode yang menggabungkan *sampling Goal Biassing*, *Gaussian*, dan *Boundary* menggunakan bahasa pemrograman Labview.

Berdasarkan permasalahan diatas, maka penulis mengangkat sebuah topik pembahasan yang akan digunakan pada pelaksanaan tugas akhir ini. Penulis mengangkat sebuah judul “Integrasi Metode *Sampling Goal Biassing*, *Gaussian*, dan *Boundary* Pada Algoritma *Rapidly Exploring Random Tree\** ”.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang, dapat dipaparkan beberapa masalah yang perlu diselesaikan, berikut beberapa permasalahannya:

- a. Saat ini telah banyak bahasa pemrograman dari algoritma dari *Rapidly Exploring Random Tree\**, namun sejauh yang penulis ketahui belum ada implementasi menggunakan bahasa pemrograman Labview
- b. Belum adanya metode pengambilan sampel yang merupakan gabungan dari metode *Goal Biassing sampling*, *Gaussian*, dan *Boundary sampling*

## 1.3 Rumusan Masalah

Berdasarkan identifikasi masalah yang ada, dapat dirumuskan beberapa poin penting yang nantinya dapat mengidentifikasi tujuan dibuatnya sistem ini:

- a. Bagaimana mengimplementasikan algoritma *Rapidly Exploring Random Tree\** pada bahasa pemrograman Labview?

- b. Bagaimana mengimplementasikan algoritma *Rapidly Exploring Random Tree\** menggunakan integrasi dari metode *sampling Goal Oriented*, *Gaussian*, dan *Boundary* pada bahasa pemrograman Labview?
- c. Bagaimana perbandingan antara algoritma *Rapidly Exploring Random Tree\** metode *sampling* integrasi dengan metode *sampling Gaussian*?
- d. Bagaimana perbandingan antara algoritma *Rapidly Exploring Random Tree\** metode *sampling* integrasi dan metode *sampling Boundary*?

#### 1.4 Tujuan

Berikut tujuan dari penelitian Integrasi Metode *Sampling Goal Biassing*, *Gaussian*, dan *Boundary* Pada Algoritma *Rapidly Exploring Random Tree\** berdasarkan rumusan masalah yang disusun:

- a. Mengimplementasikan algoritma *Rapidly Exploring Random Tree\** pada bahasa pemrograman Labview
- b. Mengimplementasikan algoritma *Rapidly Exploring Random Tree\** dengan integrasi metode *sampling Goal Oriented*, *Gaussian*, dan *Boundary* pada bahasa pemrograman Labview
- c. Dapat mengetahui perbandingan antara algoritma *Rapidly Exploring Random Tree\** metode *sampling* integrasi dengan metode *sampling Gaussian*
- d. Dapat mengetahui perbandingan antara algoritma *Rapidly Exploring Random Tree\** metode *sampling* integrasi dengan metode *sampling Boundary*

## 1.5 Batasan Masalah

Agar tidak keluar dari apa yang akan dibahas maka dibuatlah batasan masalah, berikut beberapa poinnya:

- a. Pengambilan sampel menggunakan lingkungan *obstacle* sederhana (*clutter*, *SquareField BW*, *trap*)
- b. Pengambilan sampel dilakukan masing2 10 kali percobaan pada tiap *obstacle*
- c. Pengambilan sampel membandingkan metode integrasi *sampling* dengan *Gaussian sampling*, dan *Boundary sampling*.
- d. Pengambilan sampel menggunakan bahasa pemrograman Labview