

**PELACAKAN OBJEK DINAMIS MENGGUNAKAN  
PENGUKURAN BANYAK AGEN DENGAN LATENSI  
KOMUNIKASI PADA ROBOT KONTES SEPAK BOLA BERODA**

**Laporan Tugas Akhir I**

**Disusun sebagai syarat kelulusan mata kuliah  
IF4091/Tugas Akhir I dan Seminar**

**Oleh**

**BIMO ADITYARAHMAN WIRAPUTRA**

**NIM: 13517004**



**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG**

**Desember 2020**

**PELACAKAN OBJEK DINAMIS MENGGUNAKAN  
PENGUKURAN BANYAK AGEN DENGAN  
LATENSI KOMUNIKASI PADA ROBOT KONTES  
SEPAK BOLA BERODA**

**Laporan Tugas Akhir I**

Oleh

**BIMO ADITYARAHMAN WIRAPUTRA**

**NIM: 13517004**

**Program Studi Teknik Informatika**

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, 4 Desember 2020

Mengetahui,

Pembimbing,

Dr. Judhi Santoso, M.Sc.

NIP. 196302041989031002

# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>ii</b>
<b>DAFTAR GAMBAR</b>	<b>iii</b>
<b>DAFTAR TABEL</b>	<b>iv</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
I.1 Latar Belakang	1
I.2 Rumusan Masalah	3
I.3 Tujuan	3
I.4 Batasan Masalah	4
I.5 Metodologi	4
I.6 Jadwal Pelaksanaan Tugas Akhir	5
<b>BAB II STUDI LITERATUR</b>	<b>8</b>
II.1 Teori Probabilitas	8
II.1.1 Teorema Bayes	8
II.1.2 Distribusi Gaussian	9
II.2 Estimasi Keadaan Probabilistik	10
II.2.1 Pemodelan Keadaan	10
II.2.2 Penapis Bayes	11
II.2.3 Penapis Kalman	13
II.2.4 Penapis Histogram	14
II.2.5 Penapis Partikel	15
II.3 Studi Terkait	17
<b>BAB III ANALISIS DAN PERANCANGAN</b>	<b>20</b>
III.1 Analisis Masalah	20
III.2 Lingkungan Implementasi	20
III.3 Analisis dan Desain Umum Solusi	21
<b>DAFTAR PUSTAKA</b>	<b>23</b>

## DAFTAR GAMBAR

Gambar II.1 Hubungan $X, U, Z$ . . . . .	11
Gambar III.1 Struktur modul Dagozilla . . . . .	20
Gambar III.2 Diagram alir hipotesis solusi . . . . .	22

## DAFTAR TABEL

Tabel I.1	<i>Gantt chart</i> Tugas Akhir 1 . . . . .	6
Tabel I.2	<i>Gantt chart</i> Tugas Akhir 2 . . . . .	7

# **BAB I**

## **PENDAHULUAN**

Bab ini memaparkan latar belakang yang mendasari penulisan tugas akhir ini; rumusan masalah, tujuan, dan batasan tugas akhir secara formal; sampai metodologi pengerjaan tugas akhir ini.

### **I.1 Latar Belakang**

Studi di bidang robotika merupakan salah satu bidang riset dalam lingkup intelijensi buatan yang memiliki aplikasi yang sangat luas dalam kehidupan manusia. Perkembangan robotika telah memiliki dampak yang sangat besar dalam perkembangan zaman, terutama di bidang industri manufaktur dimana robot memungkinkan pekerjaan kompleks untuk dilakukan dengan volume tinggi. Secara umum, robotika telah maupun sangat berpotensi untuk membantu menciptakan peralatan otomatis yang dapat mengerjakan pekerjaan kompleks dengan akurasi, kecepatan, ketahanan, maupun tingkat keamanan yang lebih tinggi dibandingkan dengan manusia di berbagai bidang lainnya, seperti industri pertahanan, pertambangan, sampai industri kesehatan.

Dalam studi dan pengembangan robotika, peneliti mengkaji dan mempelajari bagaimana robot dapat direkayasa untuk menyelesaikan pekerjaan yang kompleks. Lingkup studi ini pun mencakup area yang luas, dari bagaimana robot dapat berinteraksi dengan manusia sampai bagaimana robot dapat berkoordinasi dengan banyak robot lainnya dalam menyelesaikan pekerjaannya. Sebagai sistem fisik siber, robot yang baik dapat melakukan persepsi dan membaca keadaan dunia dengan akurat, melakukan komputasi dan pengambilan keputusan tingkat tinggi, dan mengeksekusi suatu aksi dengan presisi.

Salah satu inisiatif untuk membangkitkan perkembangan di bidang robotika ini adalah liga pertandingan Robocup, yang merupakan liga pertandingan terbuka untuk tim pengembang dari universitas maupun organisasi lainnya

untuk merekayasa robot-robot yang dapat ditandingkan antar tim. Diantara cabang liga yang ditandingkan dalam Robocup adalah kompetisi robot sepak bola beroda, dimana peserta mengembangkan tim robot-robot beroda yang harus berkoordinasi untuk mencetak gol di gawang lawan. Misi awal RoboCup saat didirikan adalah untuk mengumpulkan tim robot yang cukup maju untuk dapat mengalahkan tim manusia juara Piala Dunia pada tahun 2050. Inisiatif serupa juga ada di Indonesia dalam bentuk Kontes Robot Indonesia yang diselenggarakan oleh Kementrian Pendidikan dan Kebudayaan antar tim dari universitas-universitas Indonesia.

Dalam rekayasa robotika, pembuatan modul persepsi yang akurat merupakan hal yang penting karena bacaan *world state* atau keadaan dunia merupakan masukan robot untuk melakukan pengambilan keputusan dan aksi, sehingga kualitas bacaan yang buruk dapat mengakibatkan pengambilan keputusan yang salah dan membahayakan keamanan robot itu sendiri, pengguna, maupun orang lain yang mungkin terlibat. Dalam konteks robot sepak bola beroda, pelacakan posisi robot sendiri (yang disebut juga lokalisasi), posisi robot lawan, maupun posisi bola merupakan informasi yang penting yang digunakan di banyak level dari pengambilan keputusan dan strategi tim sampai fungsionalitas dasar seperti mengejar, menggiring, mengoper, dan menembak bola.

Selain pengolahan data dari sensor yang dimiliki robot sendiri, pertukaran dan pengolahan data dari robot yang lain juga merupakan hal yang penting dalam pembacaan *world state* untuk mendapatkan bacaan yang selengkap dan seakurat mungkin. Dalam kontes robot sepak bola, tidak terbacanya posisi suatu objek merupakan hal yang sering terjadi karena batasan jarak efektif sensor maupun halangan visual dari objek lain. Keterlambatan atau kegagalan penyampaian informasi dari suatu robot ke robot lain yang juga dapat terjadi karena inferensi jaringan terutama di tempat kontes yang memiliki banyak penonton mengharuskan representasi informasi yang dipertukarkan dan algoritma pengolahan informasi pada robot harus didesain dengan baik.

Telah terdapat banyak penelitian mengenai cara penggabungan bacaan keadaan dari banyak robot di dalam maupun luar konteks robot sepak bola beroda, dan ada beberapa studi modifikasi algoritma estimasi sensor dengan masukan data yang mungkin terlambat, akan tetapi masih sedikit penelitian yang membahas penggabungan bacaan keadaan dari banyak robot dalam konteks robot sepak bola beroda yang mempertimbangkan latensi di komunikasi.

Dalam tugas akhir ini, penulis meneliti dan menguji coba modul persepsi pembacaan *world state* di lingkungan robot yang dikembangkan DAGOZILLA yang merupakan tim pengembang robot sepak bola beroda di ITB yang hendak mengikuti Kontes Robotika Indonesia dan RoboCup sebagai tempat yang tepat untuk dapat mengaplikasikan dan menguji coba modul tersebut di berbagai kondisi nyata di lapangan.

## **I.2 Rumusan Masalah**

Masalah yang hendak diselesaikan pada tugas akhir ini adalah mengenai desain dan pembuatan algoritma persepsi *world state* menggunakan pertukaran informasi banyak robot dengan latensi komunikasi, spesifik pada kasus pengembangan robot sepak bola beroda di tim DAGOZILLA ITB. Untuk menyelesaikan permasalahan tersebut, masalah dibagi menjadi beberapa submasalah:

1. Bagaimana memproses data dari sensor robot itu sendiri?
2. Bagaimana merepresentasi informasi yang akan dikirimkan ke robot lain?
3. Bagaimana memproses informasi yang diterima dari robot lain yang mungkin terlambat karena latensi komunikasi?

## **I.3 Tujuan**

Tujuan dari tugas akhir ini terdiri dari beberapa poin:

1. Mendapatkan algoritma yang baik untuk memproses data dari sensor robot sendiri



2. Mendapatkan representasi yang baik untuk informasi yang akan dikirimkan ke robot lain
3. Mendapatkan algoritma yang baik untuk memproses informasi dari robot lain yang mungkin terlambat
4. Melakukan implementasi algoritma di lingkungan robot tim DAGOZILLA

#### **I.4 Batasan Masalah**

Batasan masalah yang diambil untuk membatasi lingkup penelitian di tugas akhir ini adalah:

1. *World state* yang diestimasi mencakup posisi robot anggota tim dan lawan, dan posisi bola pada saat ini di lapangan
2. Eksperimen dilakukan dengan simulasi dengan validasi menggunakan uji coba di lapangan menggunakan robot tim DAGOZILLA
3. Jenis sensor yang digunakan dan frekuensi bacaan menyesuaikan konfigurasi robot tim DAGOZILLA
4. Pemodelan *error* di bacaan sensor dan latensi komunikasi disesuaikan sedekat mungkin dengan kondisi nyata di robot tim DAGOZILLA

#### **I.5 Metodologi**

Pengerjaan tugas akhir dibagi menjadi beberapa tahapan:

1. Pemodelan lingkungan dunia nyata

Pada tahap ini, dimodelkan lingkungan dunia nyata tempat robot bekerja, termasuk distribusi *error* dari sensor dan distribusi latensi dari komunikasi.

2. Pembuatan lingkungan simulasi

Pada tahap ini, dibuat lingkungan simulasi yang dapat disambungkan dengan lingkungan kode robot untuk melakukan pengujian dengan profil *error* dan latensi sesuai yang sudah didapat sebelumnya.

### 3. Implementasi dan pengujian algoritma pembandingan

Pada tahap ini, diimplementasi dan diuji coba algoritma-algoritma dari referensi sebagai pembandingan untuk algoritma hipotesis yang akan dikembangkan.

### 4. Desain dan implementasi algoritma hipotesis

Pada tahap ini, dilakukan desain dan implementasi algoritma hipotesis di lingkungan robot agar dapat diujicobakan.

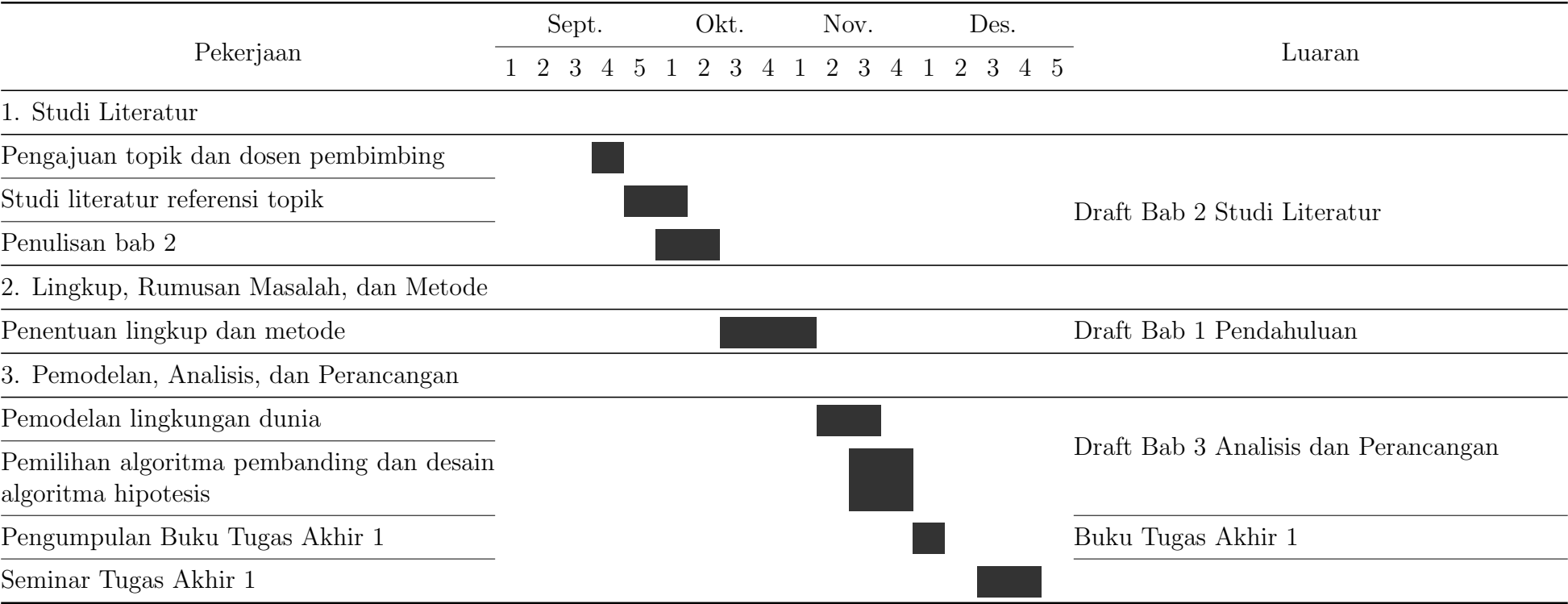
### 5. Pengujian dan analisis algoritma hipotesis

Pada tahap ini, dilakukan uji coba dan analisis dari performa algoritma hipotesis berdasarkan tujuan tugas akhir dan perbandingannya dengan algoritma pembandingan.

## **I.6 Jadwal Pelaksanaan Tugas Akhir**

Waktu pengerjaan tugas akhir ini dibagi berdasarkan diagram:

Tabel I.1. *Gantt chart* Tugas Akhir 1



2

Pekerjaan	Jan.				Feb.				Mar.					Apr.				Mei				Juni					Luaran
	1	2	3	4	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4	1	2	3	4	5	
4. Implementasi dan Pengujian																											
Implementasi lingkungan simulasi																											
Implementasi algoritma pembanding																											
Implementasi algoritma hipotesis																											
Pengujicobaan kinerja algoritma																											
5. Analisis dan Evaluasi																											
Analisis kinerja algoritma																											
Evaluasi tugas akhir																											
Seminar Tugas Akhir 2																											
Validasi tugas akhir																											
Sidang Tugas Akhir																											

## BAB II

### STUDI LITERATUR

Bab ini memberikan pengetahuan prasyarat yang mendasari isi dari tugas akhir ini. Secara spesifik, bab ini membahas dasar teori probabilitas yang digunakan, aplikasinya dalam estimasi keadaan probabilistik, dan algoritma terkait; struktur dan cara kerja modul terkait pada perangkat lunak robot sepak bola tim Dagozilla; dan studi terkait yang sudah ada pada topik estimasi keadaan multiagen maupun estimasi keadaan dengan latensi data.

#### II.1 Teori Probabilitas

##### II.1.1 Teorema Bayes

Analog dengan definisi yang ada pada probabilitas kejadian, didefinisikan juga peluang kondisional

$$p(x | y) = Pr(X = x | Y = y) = \frac{p(x, y)}{p(y)} \quad (\text{II.1})$$

yang digunakan untuk memodelkan peluang nilai  $x$  terjadi apabila nilai  $y$  terjadi. Dua variabel acak  $X$  dan  $Y$  disebut independen apabila untuk semua kemungkinan  $x$  dan  $y$  berlaku

$$p(x, y) = p(x)p(y) \quad \text{atau} \quad p(x | y) = p(x) \quad (\text{II.2})$$

Terkait peluang kondisional, Teorema Bayes menyatakan bahwa

$$p(x | y) = \frac{p(y | x)p(x)}{p(y)} = \eta p(y | x)p(x) \quad (\text{II.3})$$

dimana  $\eta = p(y)^{-1}$  merupakan suatu nilai yang konstan untuk semua kemungkinan  $x$ . Karena jumlah dari semua nilai  $p(x | y)$  haruslah bernilai satu,  $\eta$  disebut juga faktor normalisasi dan dapat ditentukan kemudian sebagai jumlah dari nilai  $p(y | x)p(x)$  untuk semua kemungkinan  $x$ .

Teorema Bayes sangat berguna untuk memperbarui kepercayaan terhadap distribusi peluang  $x$  setelah mengetahui terjadinya  $y$  apabila diketahui peluang terjadinya  $y$  dapat ditentukan untuk setiap kemungkinan  $x$ .

### II.1.2 Distribusi Gaussian

Salah satu distribusi yang penting dan banyak dipelajari adalah keluarga distribusi Gaussian atau distribusi normal dimana dengan nilai rata-rata  $\mu$  dan nilai variansi  $\sigma^2$  memiliki fungsi densitas

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left( \frac{x - \mu}{\sigma} \right)^2 \right\} \quad (\text{II.4})$$

dengan kasus spesial dimana  $\mu = 0$  dan  $\sigma^2 = 1$  disebut juga distribusi normal standar.

Distribusi Gaussian banyak dipelajari karena beberapa alasan. Salah satu alasan adalah karena distribusi ini memiliki banyak properti yang membuatnya mudah dimanipulasi dan digunakan, diantaranya adalah fungsi distribusi penjumlahan variabel-variabel acak independen berdistribusi Gaussian, maupun perkalian dan konvolusi fungsi Gaussian juga merupakan fungsi Gaussian. Alasan lainnya adalah diobservasi juga berbagai variabel acak di berbagai pengujian ilmiah yang ternyata memiliki distribusi yang hampir mirip dengan Gaussian. Selain itu juga, terbukti bahwa berdasarkan Teorema Limit Pusat, rata-rata dari banyak variabel acak distribusi identik independen akan mendekati distribusi Gaussian. (DeGroot & Schervish, 2012)

Disebabkan oleh alasan-alasan tersebut, praktisnya distribusi Gaussian banyak digunakan untuk memodelkan berbagai variabel acak di dunia nyata, seperti distribusi sampel dari populasi yang sangat besar maupun distribusi error pada suatu pengukuran atau kontrol sering diasumsikan memiliki distribusi Gaussian.

Distribusi Gaussian memiliki generalisasi ke dalam vektor acak yaitu kumpulan variabel acak yang direpresentasikan dalam bentuk vektor. Dengan parameter vektor rata-rata  $\mu$  dan matriks kovarian  $\Sigma$  yang bersifat simetrik dan positif semidefinit, didefinisikan fungsi densitas

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) \right\} \quad (\text{II.5})$$

dimana masing-masing komponen  $X_i$  sendiri memiliki distribusi Gaussian dengan rata-rata  $\mu_i$  dan kovarian antara  $X_i, X_j$  sama dengan  $\Sigma_{i,j}$ .

## II.2 Estimasi Keadaan Probabilistik

Perangkat lunak di robot atau sistem lainnya yang membaca *world state* nyata menggunakan sensor, dan mungkin berinteraksi dengannya menggunakan aktuator, mempunyai kebutuhan untuk menentukan keadaan tersebut dari bacaan sensor. Secara umum, digunakan model matematika yang bertujuan untuk mentranslasi data pengukuran menjadi model *world state*. Seiring berjalannya waktu, penggunaan estimasi *world state* menggunakan model probabilistik menjadi lebih populer dibandingkan model perhitungan yang deterministik. Dengan memodelkan *world state* yang diketahui menjadi suatu distribusi peluang, hasil pengukuran *world state* menjadi lebih tahan terhadap efek error pada data pengukuran. (Thrun dkk., 2010)

### II.2.1 Pemodelan Keadaan

Dalam model probabilistik, *world state* disimbolkan sebagai suatu vektor acak  $X$ , dimana  $x_t$  menandakan *world state* pada waktu ke- $t$ . Vektor acak ini mengandung berbagai variabel acak misalnya posisi dan orientasi sesungguhnya dari robot maupun objek lain pada suatu waktu.

Sistem melakukan pengukuran terhadap dunia yang disimbolkan sebagai vektor acak  $Z$  dan instansiasinya  $z_t$  yang merupakan pengukuran dari dunia nyata pada waktu ke- $t$ . Sedangkan apabila sistem melakukan interaksi umpan balik terhadap dunia nyata menggunakan suatu aktuator yang dikontrol oleh sistem, maka perintah interaksi tersebut disimbolkan sebagai vektor acak  $U$  dan  $u_t$  yang merupakan kontrol interaksi saat sistem sedang melakukan transisi dari  $x_{t-1}$  ke  $x_t$ .

Perhatian dari estimasi keadaan probabilistik adalah mengestimasi nilai dari  $x_t$  diberikan nilai  $z$  dan  $u$  yang terjadi sebelumnya. Dalam persamaan matematika, hendak dihitung

$$bel(x_t) = p(x_t \mid z_{1:t}, u_{1:t}), \quad (\text{II.6})$$

atau distribusi peluang alternatif dimana diprediksi nilai dari  $x_t$  sebelum mendapat hasil pengukurannya  $z_t$ ,

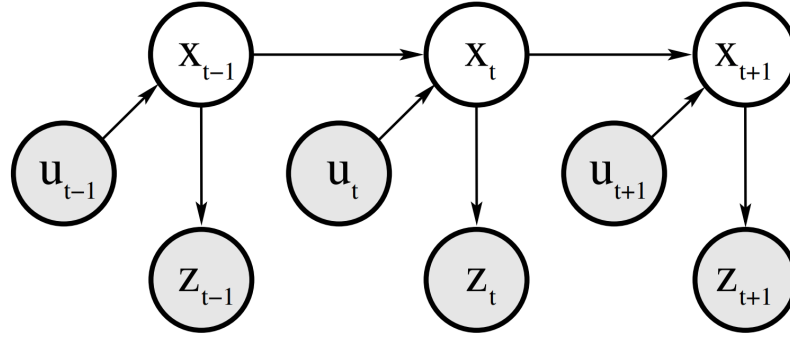
$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t}). \quad (\text{II.7})$$

Mengasumsikan nilai  $X$  mengandung semua *world state* yang relevan terhadap sistem pada waktu sekarang, maka tidak ada informasi tambahan yang bisa diberikan oleh pengukuran  $z_t$  maupun keadaan sebelumnya  $x_{t-k}$  terhadap keadaan berikutnya  $x_{t+1}$  di luar apa yang terkandung pada  $x_t$ . Oleh karena itu dalam pembangkitan nilai  $x_t$  dan  $z_t$ , dua distribusi peluang yang penting diperhatikan adalah peluang transisi keadaan

$$p(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t | x_{t-1}, u_t) \quad (\text{II.8})$$

dan peluang pengukuran

$$p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t | x_t) \quad (\text{II.9})$$



Gambar II.1. Hubungan  $X, U, Z$

Hubungan ini digambarkan pada gambar II.1. Peluang yang hanya saling berpengaruh antar tahapan waktu yang berdekatan disebut juga *hidden Markov model* atau *dynamic Bayes network*.

## II.2.2 Penapis Bayes

Perhatikan bahwa perubahan nilai dari  $\overline{bel}(x_{t-1})$  ke  $bel(x_t)$  dapat ditentukan dari peluang transisi keadaan  $p(x_t | x_{t-1}, u_t)$ , dan perubahan nilai dari  $bel(x_t)$



---

**Algoritma II.1** Penapis Bayes

---

```
1: function BAYES_FILTER( $bel(x_{t-1}), u_t, z_t$ )
2:   for all  $x_t$  do
3:      $\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$ 
4:      $bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t)$   $\triangleright \eta$  menormalisasi nilai dari  $bel(x_t)$ 
5:   end for
6:   return  $bel(x_t)$ 
7: end function
```

---

ke  $\overline{bel}(x_t)$  dapat ditentukan dari peluang pengukuran  $p(z_t | x_t)$  menggunakan Teorema Bayes. Memanfaatkan fakta tersebut, dirumuskan algoritma penapis Bayes atau *Bayes filter algorithm* pada algoritma II.1. Baris tiga menunjukkan perhitungan nilai  $bel(x_t)$  sebagai ekspansi dari penjumlahan nilai  $p(x_t, x_{t-1})$  saat diketahui  $u_t$  untuk semua kemungkinan  $x_{t-1}$ , spesifik untuk kasus kontinu. Tahap ini sering disebut tahap prediksi. Baris empat memanfaatkan teorema Bayes untuk mendapatkan distribusi  $x_t$  setelah mendapatkan informasi dari  $z_t$ . Tahap ini sering disebut tahap pembaruan pengukuran.

Algoritma penapis digunakan untuk mengiterasi nilai dari  $bel(x_t)$  seiring waktu, dan merupakan algoritma yang optimal mengasumsikan proses merupakan *hidden Markov model*. Perhatikan bahwa algoritma ini mengharuskan pemodelan sendiri nilai dari peluang transisi keadaan, peluang pengukuran, dan distribusi awal  $bel(x_0)$  yang akan digunakan.

Selain itu, karena nilai dari  $bel(x_t)$  merupakan distribusi peluang, dalam aplikasinya dibutuhkan pendekatan untuk memodelkan distribusi tersebut sehingga dapat dikomputasi oleh komputer. Pada praktisnya, distribusi peluang  $bel(x_t)$  dimodelkan sebagai distribusi dengan jenis yang dapat diparameterasikan seperti pada penapis Kalman, atau distribusi tersebut didiskritkan seperti pada penapis histogram atau penapis partikel.

### II.2.3 Penapis Kalman

Pada penapis Kalman atau *Kalman Filter* (*KF*), distribusi dari  $bel(x_t)$  dan  $\overline{bel}(x_t)$  diasumsikan memiliki distribusi Gaussian dengan parameter  $\mu$  dan  $\sigma^2$  yang dicari pada setiap iterasi algoritmanya. Agar penapis Bayes Gaussian bekerja, distribusi peluang transisi keadaan dan peluang pengukuran harus memiliki restriksi tambahan.

Pada variasi penapis Kalman yang paling sederhana, direstriksi bahwa (1) vektor acak  $x_t$  merupakan kombinasi linear dari  $x_{t-1}$ ,  $u_t$ , dan faktor error  $\epsilon_t$

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \quad (\text{II.10})$$

untuk  $x_t$  berdimensi  $n$ ,  $u_t$  berdimensi  $m$ ,  $A_t$  matriks berdimensi  $n \times n$ , dan  $B_t$  berdimensi  $n \times m$ , dan  $\epsilon_t$  vektor acak Gaussian dengan rata-rata nol dan kovariansi  $R_t$ ; (2)  $z_t$  merupakan kombinasi linear dari  $x_t$  dan faktor error  $\delta_t$

$$z_t = C_t x_t + \delta_t \quad (\text{II.11})$$

untuk  $z_t$  berdimensi  $k$ ,  $C_t$  berdimensi  $k \times n$ , dan  $\delta_t$  vektor acak Gaussian dengan rata-rata nol dan kovariansi  $Q_t$ ; dan (3) distribusi awal  $bel(x_0)$  memiliki distribusi Gaussian. Ketiga asumsi ini menjamin bahwa distribusi  $bel(x_t)$  merupakan distribusi Gaussian untuk semua waktu  $t$  yang akan datang.

---

#### Algoritma II.2 Penapis Kalman

---

- 1: **function** KALMAN\_FILTER( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ )
  - 2:      $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
  - 3:      $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
  - 4:      $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
  - 5:      $\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$
  - 6:      $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
  - 7:     **return**  $\mu_t, \Sigma_t$
  - 8: **end function**
-

Berdasarkan hubungan di atas, algoritma penapis Kalman menghitung nilai dari parameter rata-rata  $\mu_t$  dan kovariansi  $\Sigma_t$  dari distribusi Gaussian  $bel(x_t)$ . Sehingga dapat diturunkan algoritma penapis Kalman seperti pada algoritma II.2.

Disebabkan restriksi linearitas pada peluang transisi keadaan dan peluang pengukuran, algoritma penapis Kalman tidak dapat digunakan apabila restriksi linearitas tersebut tidak dipenuhi. Agar penapis Kalman dapat digunakan untuk kasus  $x_t = g(u_t, x_{t-1}) + \epsilon_t$  dan  $z_t = h(x_t) + \sigma_t$  dimana fungsi  $g$  dan  $h$  tidak linear, fungsi tersebut harus diaproksimasi terlebih dahulu agar menjadi linear.

Pada *Extended Kalman Filter*, fungsi dilinearkan menggunakan turunan parsial dari fungsi  $g$  dan  $h$ . Dengan menghitung  $G_t = g'(u_t, \mu_{t-1})$  dan  $H_t = h'(\bar{\mu}_t)$  dimana  $g'$  merupakan turunan parsial  $g$  terhadap  $x_{t-1}$ , didapatkan persamaan linear

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \quad (\text{II.12})$$

$$h(x_t) \approx h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t) \quad (\text{II.13})$$

Nilai  $A$  dan  $C$  pada penapis Kalman dapat digantikan dengan nilai  $G$  dan  $H$ .

Pada *Unscented Kalman Filter*, fungsi dilinearkan dengan mengambil sampel nilai pada fungsi  $g$  dan  $h$ , lalu membuat persamaan linear dengan melakukan regresi pada nilai masukan secara umum. Sampel diambil di rata-rata  $\mu$  dan dua titik di sekelilingnya untuk setiap dimensi yang ada.

#### II.2.4 Penapis Histogram

Pada penapis histogram, kemungkinan nilai dari  $x_t$  terbatas dan dibagi-bagi menjadi sebanyak terbatas daerah, misalnya kemungkinan posisi dari robot pada lapangan berdimensi  $9 \times 6$  meter persegi dapat dibagi menjadi 1350 daerah berbentuk persegi berukuran  $20 \times 20$  sentimeter persegi dalam suatu grid. Pada algoritma ini,  $bel(x_t)$  direpresentasikan sebagai suatu tabel yang menyimpan peluang nilai berada dalam masing-masing daerah yang ada. Algoritma ini pun relatif lebih sederhana karena hanya perlu mendiskritkan

---

**Algoritma II.3** Penapis Bayes Diskrit

---

```
1: function DISCRETE_BAYES_FILTER( $\{p_{k,t-1}\}, u_t, z_t$ )
2:   for all  $k$  do
3:      $\bar{p}_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i) p_{i,t-1}$ 
4:      $p_{k,t} = \eta p(z_t \mid X_t = x_k) \bar{p}_{k,t}$   $\triangleright$  dinormalisasi untuk semua  $k$ 
5:   end for
6:   return  $\{p_{k,t}\}$ 
7: end function
```

---

perhitungan pada filter Bayes.

Algoritma filter Bayes diskrit ada di algoritma II.3. Apabila fungsi peluang transisi keadaan atau peluang pengukuran yang dimiliki bersifat kontinu, dapat diaproksimasi menjadi diskrit seperti

$$p(\mathbf{x}_{k,t} \mid u_t, \mathbf{x}_{i,t-1}) \approx |\mathbf{x}_{k,t}| p(\hat{x}_{k,t} \mid u_t, \hat{x}_{i,t-1}) \quad (\text{II.14})$$

$$p(z_t \mid \mathbf{x}_{k,t}) \approx p(z_t \mid \hat{x}_{k,t}) \quad (\text{II.15})$$

dimana  $\hat{x}_{k,t}$  adalah representasi dari daerah  $\mathbf{x}_{k,t}$  seperti titik tengahnya, dan  $|\mathbf{x}_{k,t}|$  adalah luas daerahnya

Teknik histogram ini berkaitan erat dengan grid okupansi dimana masing-masing daerah di grid tersebut diisi dengan tingkat kepercayaan suatu nilai berada dalam daerah tersebut, walau berbeda dengan histogram peluang dimana jumlah dari nilai semua grid haruslah bernilai satu. Aplikasi dari teknik ini dalam permasalahan lokalisasi robot kerap disebut dengan *Markov localization*.

### II.2.5 Penapis Partikel

Pada penapis partikel, distribusi peluang nilai dari  $bel(x_t)$  direpresentasikan dengan menyimpan koleksi sejumlah nilai  $x_t$  atau disebut sebagai partikel, dimana distribusi dari partikel tersebut mengaproksimasi distribusi dari  $bel(x_t)$  yang sesungguhnya. Semakin banyak partikel yang disimpan, semakin akurat algoritma penapis partikel ini, tetapi semakin besar juga sumber daya memori dan waktu yang dibutuhkan.

---

**Algoritma II.4** Penapis Partikel

---

```
1: function PARTICLE_FILTER( $\chi_{t-1}, u_t, z_t$ )
2:    $\bar{\chi}_t = \bar{\chi}_t = \emptyset$ 
3:   for  $m = 1$  to  $M$  do
4:     sample  $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$ 
5:      $w_t^{[m]} = p(z_t | x_t^{[m]})$ 
6:     add  $(x_t^{[m]}, w_t^{[m]})$  to  $\bar{\chi}_t$ 
7:   end for
8:   for  $m = 1$  to  $M$  do
9:     draw  $i$  with probability  $\propto w_t^{[i]}$ 
10:    add  $x_t^{[i]}$  to  $\chi_t$ 
11:   end for
12:   return  $\chi_t$ 
13: end function
```

---

Algoritma penapis partikel digambarkan di algoritma II.4. Baris empat merupakan tahap prediksi dimana untuk setiap partikel sebelumnya dibangkitkan partikel sekarang menggunakan nilai kontrol. Partikel tersebut disimpan di himpunan  $\bar{\chi}_t$  beserta evaluasi peluangnya berdasarkan pengukuran. Agar suatu partikel dengan peluang yang lebih besar memiliki lebih banyak representasi dalam himpunan  $\chi_t$ , dilakukan sampling ulang terhadap  $x_t$  dengan peluang sebanding dengan hasil evaluasinya, pada baris delapan sampai sebelas.

Secara umum, penapis partikel ini merupakan pilihan paling populer dalam melakukan estimasi terhadap keadaan nonGaussian, karena komputasinya yang mengandalkan sampling relatif lebih mudah dan tingkat akurasi terhadap performa algoritma dapat diatur dengan mudah melalui banyak partikel yang digunakan. Ada beberapa kondisi yang mengakibatkan ketidakakuratan pada algoritma ini, seperti saat terjadi konvergensi partikel di nilai yang salah, sehingga peningkatan pada algoritma ini diantaranya adalah dengan menambahkan faktor error tambahan pada tahap prediksi dan/atau

memasukkan partikel baru di luar  $\bar{\chi}_t$  ke dalam  $\chi_t$ .

Penggunaan populer dari algoritma ini adalah pada algoritma *Monte Carlo localization* (*MCL*) dimana penapis partikel digunakan untuk kasus lokalisasi robot menggunakan data kontrol dari kontrol kecepatan atau data perpindahan dari sensor dan data pengukuran dari sensor peraba jarak atau kamera. Algoritma *Augmented Monte Carlo localization* (*AMCL*) merupakan modifikasi algoritma *MCL* biasa yang ditambahkan pemasukkan partikel baru apabila hasil evaluasi partikel-partikel yang ada sekarang lebih buruk dari sebelumnya.

### II.3 Studi Terkait

Secara umum, teknik probabilistik seperti penapis Kalman maupun penapis partikel telah banyak diimplementasi dalam kasus pelacakan objek oleh satu agen dan telah menunjukkan tingkat keberhasilan yang cukup baik. Sedangkan dalam kasus pelacakan objek oleh banyak agen, terdapat banyak studi yang mencetuskan beragam ide yang sangat bervariasi dalam berbagai konteks. Dalam konteks kontes robot sepak bola sendiri, sudah terdapat beberapa studi yang mencoba menggabungkan data pelacakan objek maupun lokalisasi.

Robot-robot Stroupe dkk., 2001 melacak posisi bola menggunakan penapis Kalman dan distribusi hasilnya disebarkan ke robot lainnya, dimana distribusi Gaussian dari semua robot akan digabungkan oleh masing-masing robot untuk menghasilkan suatu distribusi Gaussian akhir. Asumsi dasar dari makalah ini adalah *error*-nya bersifat normal independen, lokalisasi robot sempurna, dan waktu pengukuran bersamaan.

Ferrein dkk., 2005 membandingkan beberapa metode untuk menggabungkan observasi bola dari masing-masing robot, seperti merata-ratakan posisi pengamatan, menggunakan penapis Kalman menggunakan terhadap posisi bola menggunakan data yang didapat masing-masing robot, menggunakan penapis partikel, maupun menggunakan penapis histogram yang digabungkan ke estimasi global menggunakan penapis Kalman. Hasilnya adalah penapis Kalman sederhana memiliki tingkat akurasi dan kecepatan yang paling baik.

Masing-masing robot Pahliani & P. Lima, 2006 melakukan lokalisasi menggunakan *Markov localization*, lalu tergantung dari posisi robot, robot-robot yang ada membentuk beberapa kelompok kecil untuk bertukar informasi. Data dari robot di kelompok yang sama dibagikan, lalu menggunakan data yang didapat dari kelompok lain, hasil lokalisasi dari masing-masing robot diperbaiki secara Bayes. Setelah lokalisasi, deteksi objek juga dilakukan menggunakan alur dan kelompok yang sama.

Santos & P. Lima, 2009 melacak bola menggunakan penapis partikel. Untuk mengefisiensikan pertukaran informasi antar robot, distribusi dalam bentuk partikel diubah dulu menjadi *gaussian mixture model* yang terdiri dari beberapa distribusi Gauss menggunakan algoritma *expectation maximization* yang bekerja mirip seperti algoritma *K means*. Ada juga ukuran persetujuan informasi antar robot menggunakan *covariance intersection* untuk menentukan apakah informasi baru diintegrasikan dengan informasi robot sendiri atau tidak. Ahmad & P. U. Lima, 2011 mengembangkan ide ini dengan membuat algoritma penggabungan distribusi dari masing-masing robot dimana dibangkitkan kumpulan partikel baru dengan melakukan *sampling* dari distribusi-distribusi yang ada berdasarkan tingkat kepercayaan distribusi tersebut sebagai masukan dari penapis partikel.

Ahmad, Tipaldi, dkk., 2013 mengumpulkan semua pengukuran posisi relatif semua robot terhadap robot lain, objek statis, maupun objek dinamis, dan mencari konfigurasi posisi semua objek yang meminimasi *error* dari pengukuran-pengukuran yang ada menggunakan *iterative local linearization* atau *least square minimization*. Kelebihan teknik ini diantara adalah beban komputasinya linear terhadap banyak robot, akan tetapi setiap robot harus menunggu sampainya data pengukuran dari semua robot lain.

Chang dkk., 2016 menggunakan *Extended Kalman Filter* untuk melacak sistem seluruh robot dan objek. *Update* dilakukan untuk masing-masing robot, sedangkan pengukuran diintegrasikan berdasarkan robot apa saja yang terlibat dalam pengukuran tersebut. Teknik *multiple hypothesis tracking* digunakan

untuk mengasosiasikan data pengukuran dengan posisi objek di hipotesis.

Ahmad, Lawless, dkk., 2017 menggunakan penapis partikel yang mencakup posisi semua robot ditambah objek yang diobservasi. Masing-masing bagian partikel dari robot digerakan dan dievaluasi secara terpisah, dan diurutkan ulang sehingga bagian partikel dengan evaluasi tinggi dari suatu robot dipasangkan dengan bagian partikel robot lain dengan evaluasi yang tinggi juga. Algoritma ini lalu memasangkan partikel bagian dari objek yang memaksimalkan evaluasi partikel bagian tersebut dengan partikel bagian robot dengan evaluasi tinggi juga. Teknik ini menyelesaikan masalah defisiensi partikel pada penapis partikel tanpa harus menambahkan banyak partikel secara eksponensial.



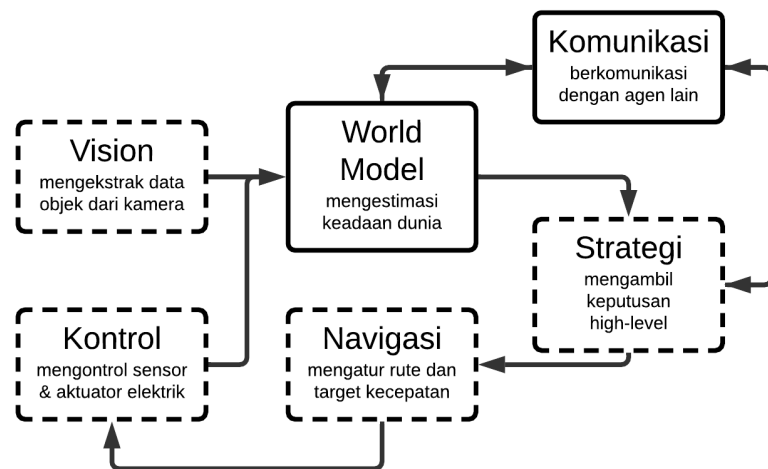
## BAB III

### ANALISIS DAN PERANCANGAN

#### III.1 Analisis Masalah

Hampir semua studi pengembangan algoritma estimasi *world model* yang melacak posisi robot maupun bola di kontes robot sepak bola mengasumsikan bahwa komunikasi antar robot memiliki latensi yang dapat diabaikan dan belum ada yang mencoba menangani secara eksplisit dan menganalisis pengintegrasian data dari robot lain yang terlambat. Secara umum, masih sedikit studi yang membahas penggabungan data pengukuran sensor dengan latensi pada sistem komputasi terdesentralisasi.

#### III.2 Lingkungan Implementasi



Gambar III.1. Struktur modul Dagozilla

Lingkungan perangkat lunak DAGOZILLA menggunakan *platform Robot Operating System (ROS)* yang memungkinkan komunikasi antar proses oleh modul-modul aplikasi yang menjalankan fungsi-fungsi robot menggunakan paradigma *publish-subscribe* pesan atau *request-response service* yang

disediakan suatu modul. Modul-modul yang ada digambarkan pada gambar III.1.

Fokus utama dari tugas akhir ini adalah mengembangkan modul *world model* yang bertugas untuk memproses semua data persepsi dari modul sebelumnya maupun informasi dari robot lain untuk menyediakan estimasi keadaan dunia yang seakurat dan sefaktual mungkin agar dapat digunakan dalam melakukan pengambilan keputusan.

Modul kontrol berkomunikasi dengan mikrokontroler yang berhubungan langsung dengan *hardware* untuk mengontrol aktuator dan mengambil data dari sensor-sensor lokal, diantaranya odometri motor penggerak dan kompas. Odometri mengukur perputaran yang dilakukan masing-masing roda penggerak yang dapat diolah menjadi data perpindahan dan kompas mengukur orientasi global dari robot itu sendiri.

Modul *vision* mengekstrak data dari kamera, diantaranya posisi bola dari gambar menggunakan deteksi warna dan momen objek, dan sebaran titik dari garis lapangan maupun objek penghalang lapangan menggunakan algoritma *radial search line*.

### III.3 Analisis dan Desain Umum Solusi

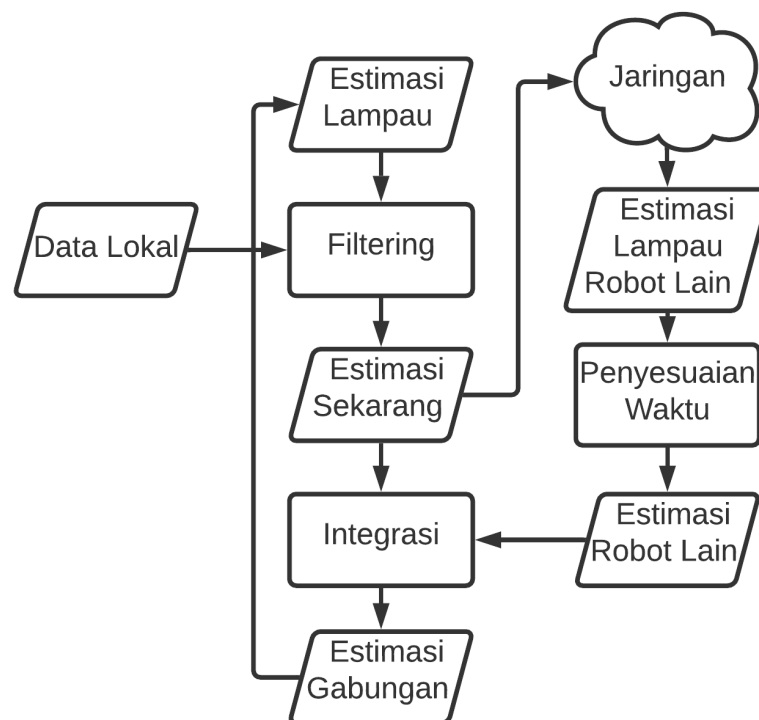
Sebagai gambaran besar hipotesis solusi, masing-masing robot tetap harus melakukan estimasi sendiri menggunakan data yang tersedia dari sensor lokal robot tersebut. Hal ini dilakukan agar masing-masing robot harus tetap memiliki estimasi informasi krusial seperti posisi robot sendiri, bola, ataupun penghalang walaupun terjadi kegagalan jaringan komunikasi. Algoritma secara umum harus berjalan secara *online* tanpa masing-masing robot harus menunggu informasi dari robot lain terlebih dahulu sebelum dapat dijalankan. Penapis partikel dianggap sebagai metode yang cukup baik untuk melacak posisi robot maupun bola yang dapat bergerak secara bebas.

Masing-masing robot lalu menyiarkan hasil pemrosesan lokalnya ke semua robot lain. Mengirimkan keseluruhan informasi pengukuran secara mentah

dianggap kurang dapat diandalkan dan meningkatkan beban komputasi untuk semua robot melihat ada data seperti persepsi garis lapangan dan objek penghalang yang terdiri dari puluhan sampai ratusan data titik. Data seperti sebaran partikel estimasi haruslah diproses dahulu ke bentuk lain seperti *Gaussian Mixture Model* sebelum disiarkan ke robot lain.

Data yang disiarkan mengandung *time-stamp* pengolahan data agar robot yang menerima dapat mendeteksi dan melakukan pengolahan tambahan untuk data yang terlambat. Data juga mengandung ukuran tingkat kepercayaan keakuratan yang ditetapkan robot penyiar. Hal ini dilakukan karena walaupun pertukaran informasi dapat meningkatkan akurasi, hal tersebut juga dapat menyebabkan penyebaran *error* dalam estimasi, sehingga harus didesain sehingga informasi dengan tingkat kepercayaan yang rendah tidak terlalu mempengaruhi informasi dengan tingkat kepercayaan yang lebih tinggi.

Diagram alir hipotesis solusi digambarkan di III.2.



Gambar III.2. Diagram alir hipotesis solusi

## DAFTAR PUSTAKA

- Ahmad, A., Lawless, G., dkk. (2017). An online scalable approach to unified multirobot cooperative localization and object tracking. *IEEE transactions on robotics* 33(5), pp. 1184–1199.
- Ahmad, A. & Lima, P. U. (2011). Multi-Robot Cooperative Object Tracking Based on Particle Filters. *ECMR*. Citeseer, pp. 37–42.
- Ahmad, A., Tipaldi, G. D., dkk. (2013). Cooperative robot localization and target tracking based on least squares minimization. *2013 IEEE International Conference on Robotics and Automation*. IEEE, pp. 5696–5701.
- Chang, C. dkk. (2016). Exploiting Moving Objects: Multi-Robot Simultaneous Localization and Tracking. Vol. 13. 2, pp. 810–827.
- DeGroot, M. H. & Schervish, M. J. (2012). *Probability and statistics*. Pearson Education.
- Ferrein, A. dkk. (2005). Comparing sensor fusion techniques for ball position estimation. *Robot Soccer World Cup*. Springer, pp. 154–165.
- Pahliani, A. & Lima, P. (2006). Improving Self Localization and Object Localization by a Team of Robots Using Sensor Fusion. *Proc. of CONTROLO*.
- Santos, J. & Lima, P. (2009). Multi-robot cooperative object localization. *Robot Soccer World Cup*. Springer, pp. 332–343.
- Stroupe, A. W. dkk. (2001). Distributed sensor fusion for object position estimation by multi-robot systems. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*. Vol. 2, 1092–1098 vol.2.
- Thrun, S. dkk. (2010). *Probabilistic robotics*. MIT Press.