BUKU RANCANGAN PEMBELAJARAN

ROBOTIKA

Disusun oleh:

Prof. Dr. Eng. Wisnu Jatmiko
Muhammad Anwar Ma'sum M.Kom.

Program Studi Ilmu Komputer
Fakultas Ilmu Komputer
Universitas Indonesia
2018

DAFTAR ISI

PENGANTAR

Universitas Indonesia memiliki komitmen yang kuat untuk mengalihkan paradigmanya dari pemelajaran yang berpusat pada pengajar (*teacher-center learning*) menuju pemelajaran yang berorientasi pada peserta didik/pemelajar (*student-center learning*). Pada paradigma yang baru, tanggung jawab pemelajaran berpusat pada pemelajar, sedangkan pengajar lebih banyak berperan sebagai fasilitator, *coach*, dan model. Guna menjamin keberhasilan pengajar dalam melaksanakan tugasnya diperlukan perancangan pemelajaran yang jelas, mampu terlaksana, dan mengacu pada tujuan serta sasaran pemelajaran dengan tidak lupa memperhatikan karakteristik pemelajar.

Buku Rancangan Pengajaran (BRP) untuk mata ajar Robotika ini merupakan dokumentasi dari rancangan pemelajaran yang bersifat menyeluruh. Buku ini diharapkan dapat digunakan sebagai acuan bagi pengajar atau tim pengajar sehingga memudahkan koordinasi dalam pemelajaran. BRP ini diharapkan dapat semakin disempurnakan sesuai dengan perkembangan kurikulum dan bahan pemelajaran.

Dalam konstelasi kajian Ilmu Komputer, Robotika memberikan pengantar tentang implementasi robot dari segi *software* termasuk kendali dan kecerdasan buatan. Topik yang dibahas pada kuliah ini meliputi pergerakan di tempat, perpindahan, penagkapan informasi dari luar, lokalisasi, pemetaan, dan navigasi. Pada kuliah ini, mahasiswa juga akan melakukan banyak praktikum menggunakan robot nyata. Robot tersebut dapat diprogram dengan menggunakan berbagai algoritma pembelajaran mesin maupun kecardasan buatan. Hal tersebut akan dapat membuat pemelajar memiliki ketangkasan maupun pemahaman menyeluruh tentang dunia robotika.

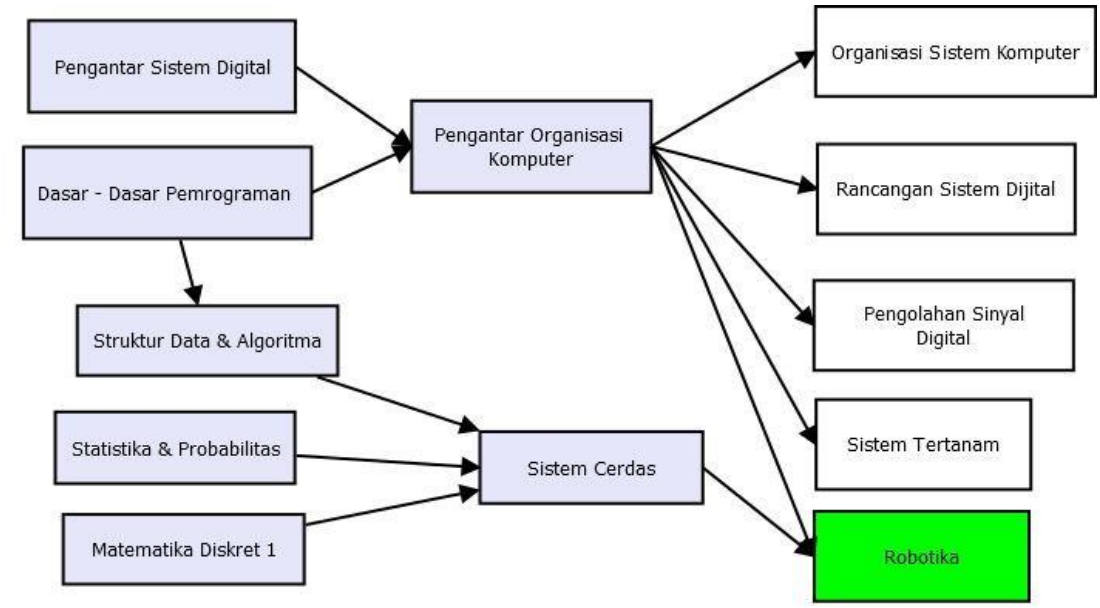
Januari 2018

Dr. Eng. Wisnu Jatmiko

BAB 1
INFORMASI UMUM

1) Nama Program Studi/ jenjang : Ilmu Komputer / Sarjana
2) Nama mata kuliah/topik : Robotika
3) Kode Topik/modul/mata kuliah : IKO42360
4) Semester : 6
5) Jumlah SKS : 3
6) Metode pembelajaran : Tutorial, Student Center Learning (Praktikum), dan *on line learning* melalui SCELE
7) Mata kuliah/modul yang menjadi prasyarat : Pengantar Organisasi Komputer dan Sistem Cerdas
8) Menjadi prasyarat kuliah : Tidak ada
9) Integrasi/kaitan antara mata kuliah :

Mata ajar Robotika merupakan kuliah lanjutan rumpun ilmu arsirtektur komputer dan kecerdasan komputasional. Syarat untuk dapat mengambil mata kuliah ini adalah pernah mengambil mata kuliah Pengantar Organisasi Komputer dan Sistem Cerdas. Keterkaitan antar mata kuliah ini dapat dilihat pada bagan di bawah ini.



10) Deskripsi mata kuliah/modul :

Mata ajar Robotika memberikan pengantar tentang implementasi robot dari sudut pandang ilmu komputer. Topik yang dibahas pada kuliah ini meliputi pergerakan di tempat, perpindahan, penagkapan informasi dari luar, lokalisasi, pemetaan, dan navigasi. Pada kuliah ini, mahasiswa akan melakukan praktikum menggunakan simulasi maupun robot nyata. Robot tersebut dapat diprogram dengan menggunakan berbagai algoritma pembelajaran mesin maupun kecardasan buatan. Hal tersebut akan dapat membuat pemelajar memiliki ketangkasan maupun pemahaman menyeluruh tentang dunia robotika dalam riset maupun industri.

BAB 2

SASARAN PEMBELAJARAN

2.1. Sasaran Pembelajaran Terminal

Setelah menyelesaikan modul Kuliah Robotika, peserta kuliah diharapkan mampu melakukan implementasi algoritma-algoritma kecerdasan buatan maupun pembelajaran mesin pada platform robot apapun. Hal tersebut disebabkan walau robot manapun yang digunakan, konsep dari ilmu yang telah dipelajari tetaplah sama. Hal ini juga didukung oleh banyaknya ketersediaan robot beserta *software development kit* (SDK) yang ada di pasaran. Mahasiswa yang lulus mata kuliah ini juga diharapkan memiliki daya saing dalam dunia kerja maupun penelitian yang terkait aplikasi pada robot.

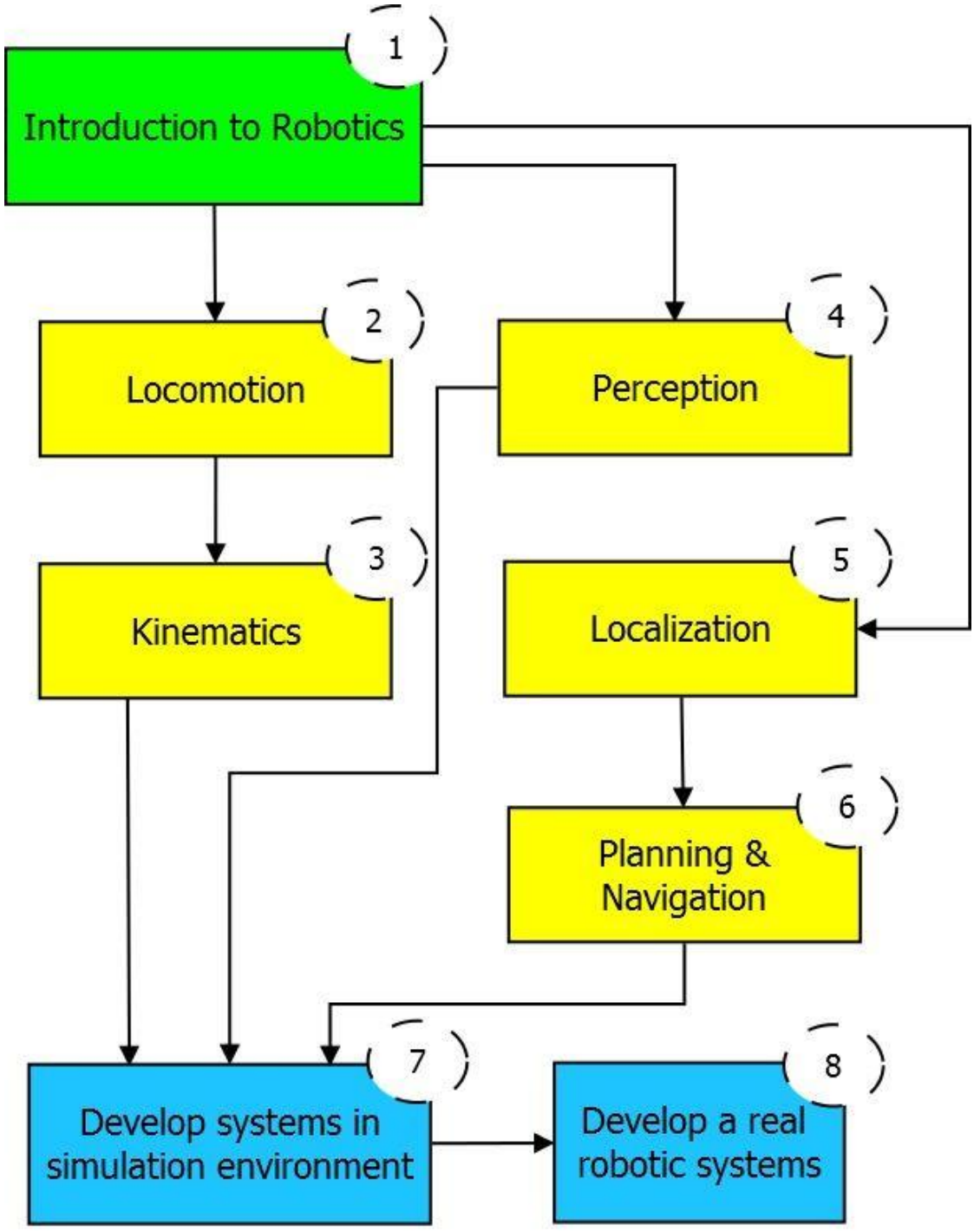
2.2. Sasaran Pembelajaran Penunjang

2.2.1. Jika dihadapkan dengan perangkat maupun sistem robot, mahasiswa mampu (C3):
   a. mengoprasikan peralatan dengan baik.
   b. menjelaskan bagaimana peralatan tersebut bekerja secara teknis.
   c. mengetahui suatu aplikasi berjalan baik atau tidak saat dioperasikan.

2.2.2. Jika dihadapkan pada kebutuhan *framework* yang digunakan dalam pengembangan robot, mahasiswa mampu (C4P3A3):
   a. membuat daftar perangkat keras maupun lunak yang dibutuhkan.
   b. melakukan instalasi dan integrasi sistem yang akan digunakan untuk pengembangan.
   c. menjalankan dan memahami cara kerja aplikasi sederhana atau aplikasi demo pada *framework* tersebut.
   d. menilai hubungan antar komponen, perangkat keras maupun lunak, pada sistem yang berjalan.

2.2.3. Jika dihadapkan pada kebutuhan implementasi sistem robot secara keseluruhan, mahasiswa mampu (C5P3A3):
   a. menetapkan misi yang akan dilakukan oleh robot.
   b. membuat desain sistem secara keseluruhan sesuai kebutuhan (misi yang telah ditetapkan).
   c. membuat kontrol terhadap arah gerak robot.
   d. mengatur perilaku-perilaku yang dilakukan robot selama menjalankan misi.
   e. mengolah data yang diterima oleh sensor sebagai informasi yang bermanfaat untuk membuat keputusan (*decision*).
   f. menguji sistem yang dibangun apakah bekerja sesuai desain yang dibuat di awal.

2.3. Bagan Alir Sasaran Pembelajaran



Keterangan :

Konsep dasar

Konsep lanjut

Mengimplementasikan konsep

BAB 3
BAHASAN DAN RUJUKAN

3.1. Bahasan

No	Pokok Bahasan	Subpokok bahasan	Estimasi waktu	Rujukan
1	Introduction: problem statements, typical applications, video	1.1 Introduction and Videos 1.2 Example, History, and Challenge	3 x 50 Menit	[1] chap 1 hal 1-12 [2] Slide di SCeLE
2	Locomotion	2.1 Introduction of Locomotion Mechanism 2.2 Legged Mobile Robots 2.3 Wheeled Mobile Robots	6 x 50 Menit	[1] chap 2 hal 13-17 [2] chap 2 hal 17-30 [3] chap 2 hal 30-45 [4] Slide di SCeLE
3	Kinematics	3.1 Introduction of Kinematics 3.2 Kinematic Models and Constraints 3.3 Maneuverability and Motion Control	6 x 50 Menit	[1] chap 3 hal 47-66 [2] chap 3 hal 67-88 [3] Slide di SCeLE
4	Perception	4.1 Sensors 4.2 Uncertainty Representation 4.3 Feature Extraction	8 x 50 Menit	[1] chap 4 hal 89-145 [2] chap 4 hal 145-151 [3] chap 4 hal 151-180 [4] Slide di SceLE
5	Localization	5.1 Intro Localization and Odometry 5.2 Map Representation 5.3 Probabilistic Map-Based Localization	8 x50 Menit	[1] chap 5 hal 181-200 [2] chap 5 hal 200-212 [3] chap 5 hal 212-256 [4] Slide di SCeLE
6	Planning and Navigation	6.1 Intro Planning and Navigation 6.2 Path Planning and Obstacle Avoidance 6.3 Navigation Architectures	6 x 50 Menit	[1] chap 6 hal 257-290 [2] chap 6 hal 291-304 [3] Slide di SCeLE

7	Develop robot system in simulation environment	7.1 Work on Laboratory	8 x 50 Menit [Praktikum]	Task and tutorial modules
8	Develop robot system in real hardware	8.1 Work on Laboratory	9 x 50 Menit [Praktikum]	Task and tutorial modules

3.2. Daftar Rujukan

Utama:

Roland Siegwart and Illah R. Nourbakhsh (2004). *Introduction to Autonomous Mobile Robots*. Cambridge, MA: MIT Press.

Rujukan Tambahan:

1) Wisnu Jatmiko dkk. (2010). *Robot Lego Mindstrom: Teori dan Praktek*. Jakarta: UI Press.
2) Wisnu Jatmiko dkk. (2010). *Swarm Robot dalam Pencarian Sumber Asap*. Jakarta: UI Press.
3) Wisnu Jatmiko dkk. (2012). *Robotika: Teori dan Aplikasi*. Jakarta: UI Press.

BAB 4

TAHAP PEMBELAJARAN

Media Instruksional

1. SCeLE (Sc)
2. *White board* (Wb)
3. Komputer dan LCD Proyektor (KL)
4. Komponen perangkat keras / *hardware* (Hw)
5. TM (Tugas Mandiri)
6. TK (Tugas Kelompok)

Sasaran Pembelajaran/ Sasaran Pembelajaran Penunjang	Tahap Pembelajaran**			Media Teknologi
	O (%)	L (%)	U (%)	
2.2.1.	Pengantar dan tugas mandiri (70%)	Diskusi kelompok (10%)	Pleno &umpan balik (20%)	Sc, Wb, KL, TM
2.2.2.	Pengantar dan tugas mandiri (30%)	Diskusi kelompok (40%)	Pleno & umpan balik (30%)	Sc, Wb, KL, TM, TK
2.2.3.	Pengantar dan tugas mandiri (10%)	Diskusi kelompok (60%)	Pleno & umpan balik (30%)	Sc, Wb, KL, Hw, TK

Seluruh proses pemelajaran didukung oleh media online yaitu Student Center eLearning (SCeLE). Semua mahasiswa peserta mata ajar Robotika diharapkan meng-*enroll* halaman kuliah "[REG] Robotika - Genap 2013/2014" sehingga dapat mengakses materi-materi yang tersedia di SCeLE. Materi-materi tersebut akan tersedia sebelum kuliah dimulai. Pengumuman dan tugas juga akan ditampilkan melalui SCeLE. Selain itu, mahasiswa juga dapat berdiskusi secara *online* melalui forum yang tersedia pada halaman mata kuliah. Mahasiswa dapat mengajukan pertanyaan atau merespon satu sama lain tanpa moderator. Dosen dan asisten juga dapat memonitor aktivitas diskusi dan turut merespon agar diskusi bisa lebih terarah. Jika diperlukan, dosen juga dapat memberikan pemicu berupa pertanyaan maupun permasalahan nyata yang harus dicari solusinya.

Untuk tugas akan disertakan perangkat keras penunjang berupa robot yang sudah dapat langsung digunakan. Perangkat tersebut disiapkan sesuai deskripsi tugas yang dikeluarkan.

BAB 5
RANCANGAN TUGAS DAN LATIHAN

5.1. Tujuan Tugas

Tabel uraian tugas

Sasaran Pembelajaran/ Sasaran Pembelajaran Penunjang	Objek garapan	Ruang Lingkup	Cara pengerjaan	Batas waktu	Luaran tugas yang dihasilkan
2.2.1 – 2.2.2	Pengenalan robotika dan teori dalam robotika	Teori	Di kelas, diskusi kelompok, pleno	2x2 jam	*Power point presentasi mahasiswa*, hasil presentasi, dan lembar tugas mandiri
2.2.2 – 2.2.3	Robot Simulation	Teori dan Praktek (2 kali)	Di kelas, diskusi kelompok, di laboratorium (praktikum), pleno	2x2 jam [Teori] 2x2 jam [Praktikum]	*Power point presentasi mahasiswa*, hasil presentasi, Tugas praktikum dan Demo *software* hasil simulasi
2.2.2 – 2.2.3	Real Robot Implementation	Teori dan Praktek (2 kali)	Di kelas, diskusi kelompok, di laboratorium (praktikum), pleno	2x2 jam [Teori] 2x2 jam [Praktikum]	*Power point presentasi mahasiswa*, hasil presentasi,Tugas praktikumdan Demo video hasil praktek

5.2. Kriteria Penilaian

Tugas yang diberikan terdiri dari dua bentuk, yaitu tugas teori / kuis dan tugas praktek. Permasalahan yang diberikan disesuaikan dengan laju materi pemelajaran dan merupakan latihan bagi mahasiswa untuk menerapkan ilmu yang telah diperoleh. Penguasaan teori-teori dasar biasanya tidak diujikan secara langsung, namun mahasiswa dihadapkan pada persoalan riil yang harus dicari solusinya dengan mengintegrasikan beberapa penguasaan konsep dan ketrampilan analisa maupun sintesa. Contoh tugas dan latihan dapat dilihat di lampiran.

.

BAB 6

EVALUASI HASIL PEMBELAJARAN

6.1. Evaluasi Akhir (*Tentative*)

Bentuk	Instrumen	Frekuensi	Bobot (%)
Tutorial	Modul tutorial	2	10
Tugas Mandiri	Tugas membuat tulisan	2	30
Tugas Kelompok	Implementasi program	2	50
Kuis	Soal essay	4	10
Diskusi di SCeLE	Memberi pendapat atau menjawab pertanyaan	1	Bonus
Total			**100**

6.2. Asesmen

Sasaran Pembelajaran/ Sasaran Pembelajaran Penunjang	Ranah dan tingkatan	Jenis asesmen	Indikator keberhasilan
2.2.1.	C3	- Kuis (essai) - Diskusi di SCeLE - Tugas Individu - Tugas Kelompok	Nilai minimal mencapai 70 (B)
2.2.2.	C4P3A3		
2.2.3.	C5P3A3		

6.3. Pedoman Kriteria Penilaian

Huruf	Rentang Nilai	Bobot
A	85 – 100	4.0
A-	80 – 84.99	3.7
B+	75 – 79.99	3.3
B	70 – 74.99	3.0
B-	65 – 69,99	2.7
C+	60 – 64.99	2.3
C	55 – 59,99	2.0
C-	50 – 54.99	1.7
D	40 – 49.99	1.0
E	0 – 39,99	0

/*Batas Lulus B*/

BAB 7

MATRIKS KEGIATAN

Metode/pendekatan pemelajaran:

1. Diskusi Kelompok (PBL = Problem Based Learning, CL= Collaborative Learning)
2. Diskusi Interaktif via SCeLE (DI)
3. Belajar Mandiri (BM)
4. Kuliah Interaktif / tatap muka (KI), dengan sarana slide presentasi
5. Praktikum (P)

Sumber Pemelajaran

1. Buku Teks
2. Forum diskusi SCeLE
3. *Handout presentation*
4. Video terkait robotika
5. Panduan Robot Operating System (ROS)
6. Modul Praktikum
7. Sumber lain dari internet

Pertemuan ke-	Sasaran Pembelajaran/ Sasaran Pembelajaran Penunjang	Tahap Pembelajaran			Pokok Bahasan/ SPB	Media Teknologi	Ranah dan Tingkatan	Kriteria Penilaian (Indikator)	Penanggung Jawab
		O (%)	L (%)	U (%)					
1	2.2.1	KI	BM, PBL, CL	DI	1.1-1.2	Sc,Wb,KL	C3	Nilai ≥ 70	Narasumber, fasilitator
2	2.2.1	KI	BM	DI	2.1	Sc,Wb,KL	C3	Nilai ≥ 70	Narasumber, fasilitator
3	2.2.2	KI	BM, PBL, CL	DI	2.2-2.3	Sc, Wb, KL, TM	C4P3A3	Nilai ≥ 70	Narasumber, fasilitator
4	2.2.1	KI	BM	DI	3.1	Sc,Wb,KL	C3	Nilai ≥ 70	Narasumber, fasilitator
5	2.2.2	KI	BM, PBL, CL	DI	3.1-3.2	Sc, Wb, KL, TM	C4P3A3	Nilai ≥ 70	Narasumber, fasilitator
6	2.2.1	KI	BM	DI	4.1	Sc, Wb, KL, TM	C4P3A3	Nilai ≥ 70	Narasumber, fasilitator
7	2.2.2	KI	BM, PBL, CL	DI	4.2-4.3	Sc, Wb, KL, TM	C4P3A3	Nilai ≥ 70	Narasumber, fasilitator
8	2.2.1	KI	BM	DI	5.1	Sc, Wb, KL	C3	Nilai ≥ 70	Narasumber, fasilitator
9	2.2.2	KI	BM, PBL, CL	DI	5.2-5.3	Sc, Wb, KL, TM	C4P3A3	Nilai ≥ 70	Narasumber, fasilitator

10	2.2.1	KI	BM	DI	6.1	Sc, Wb, KL	C3	Nilai ≥ 70	Narasumber, fasilitator
11	2.2.2	KI	BM, PBL, CL	DI	6.2-6.3	Sc, Wb, KL, TM	C4P3A3	Nilai ≥ 70	Narasumber, fasilitator
12	2.2.3	KI	BM, PBL, CL	DI, P	7.1	Sc, Wb, KL, TK	C5P3A3	Nilai ≥ 70	Narasumber, fasilitator
13	2.2.3	KI	BM, PBL, CL	DI, P	8.1	Sc, Wb, KL, TK	C5P3A3	Nilai ≥ 70	Narasumber, fasilitator

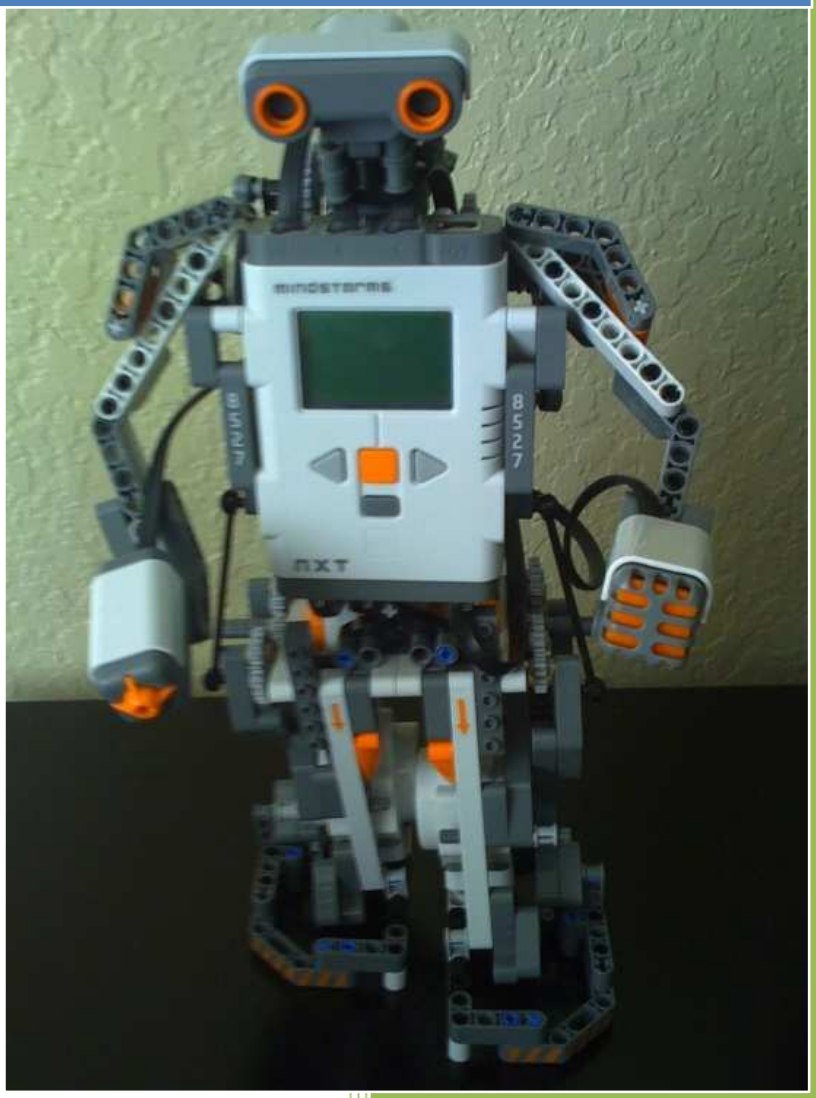
LAMPIRAN

Contoh Tugas UAS dan UTS Kuliah-kuliah Robotika Sebelumnya

Draft Proyek Pengganti UAS

Kuliah Robotika 2011

Robot “Teman Ngobrol”



Mohamad Sani

Rizki Mandala Putra

Tembra Variantoro

Daftar Isi

1 LATAR BELAKANG

Pada Tugas 1 kelompok kami telah berhasil membuat Robot Kecoa dan pada Tugas Pengganti UTS sebelumnya kami telah mengembangkan Robot Kalajengking. Pada Proyek kali ini, kami hendak beralih dari robot yang mengimitasi binatang menuju robot yag mengimitasi manusia. Tujuannya adalah agar diperoleh Robot yang lebih dekat dengan manusia sebagai pengguna robot itu sendiri. Kali ini kami akan membuat model robot Alpha Rex, robot humanoid yang berjalan dengan dua kaki. Imitasi manusia ini dilakukan dengan tujuan agar skenario sesuai dengan apa yang akan diteliti. Penelitian yang akan kami lakukan kali ini bukan efisiensi, optmisas atau analisa algoritma seperti kelompok lain, tetapi penelitian yang bertujuan ingin mengetahui seberapa jauh kemampuan instrumen robot LEGO yang tersedia lebih khususnya sensor suara bawaan LEGO dan LeJOS sebagai *programming interface*-nya. Uji coba kapabilitas sensor suara ini menarik apabila dibalut dengan skenario Robot Teman Ngobrol yang akan dibangun. kami Harapan kami riset ini dapat membuat kami memahami lebih baik teori yang telah diajarkan tersebut dengan mempraktikkannya langsung di laboratorium.

1.1. Permasalahan Penelitian

Adapun permasalahan penelitian ini adalah sebagai berikut:

1. Seberapa besar nilai:
    a. Sensitivitas (Sensitivity)
    b. Akurasi (Error/Accuracy)
    c. Presisi (Precision)

    sensor suara bawaan LEGO yang dikembangkan dengan LeJOS?
2. Seberapa tinggi kemampuan sistem robotika LEGO mengenal percakapan?

1.2. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mendapatkan nilai Sensitivitas (*Sensitivity*), Akurasi (*Error/Accuracy*), dan Presisi (*Precision*) sensor suara bawaan LEGO sehingga berguna sebagai referensi peneliti/praktisi lain yang ingin memanfaatkan sensor suara LEGO.

2. Mengetahui kemampuan LEGO dalam pengenalan percakapan.
3. Mencari cara/teknik agar robot LEGO dapat mengenali percakapan.

1.3 Skenario Narasi

Alpha Rex adalah robot yang bisa berjalan dengan dua kaki. Ia akan berjalan. Apabila ia bertemu dengan seseorang, ia akan berhenti dan menyapanya. Selama orang tersebut tidak pergi dari hadapan Alpha Rex, Alpha Rex akan menjadi teman ngobrolnya. Alpha Rex dapat membedakan apakah ia berbicara dengan orang negro, orang Indonesia atau orang bule. Ia juga merespon apabila dijabat tangannya. Alpha Rex juga akan bereaksi terhadap cahaya. Jika lampu dimatikan ia akan mengucapkan selamat malam dan tidur. Jika lampu dihidupkan lagi Alpha Rex akan menyapa selamat pagi. Yang paling menarik adalah bagaimana Alpha Rex dapat merespon percakapan lawan bicaranya…

1.4. Skenario Teknis

Begitu dihidupkan, Alpha Rex akan mulai berjalan menelusuri tempatnya berada. Alpha Rex dapat berjalan maju dan berbelok. Jika Alpha Rex menemui seseorang dihadapannya (lebih dekat dari 30 cm), ia akan menyapa. Alpha Rex menyapa sesuai dengan nilai grayscale yang dideteksinya di depannya. Alpha Rex akan menyapa dengan salam yang berbeda untuk benda gelap, sedang dan terang. Selagi orang di depannya belum beranjak dari jarak 30 cm, Alpa Rex akan mencoba merespon pembicaraan lawan bicaranya. Apabila sang lawan bicara sudah beranjak Alpha Rex akan kembali berjalan.

Alpha Rex dapat merespon jabatan tangan. Begitu pula terhadap cahaya lampu. Jika lampu dimatikan maka Alpha Rex akan mengeluarkan suara selamat malam dan tidur, jika lampu dihidupkan kembali ia akan terbangun dan menyapa selamat pagi.

2 STUDI LITERATUR

Pembangunan robot dan kode yang membuat sistem robot yang lebih efektif dan efisien. Teori yang kami gunakan mengacu dari buku *Introduction to Autonomous Mobile Robots* (Siegwart & Nourbakhsh, 2004) oleh Roland Siegwart dan Illah R. Nourbakhsh .

2.1 Kinerja Sensor In-Situ

Sensitivitas adalah perbandingan antara perubahan yang terjadi pada masukan dengan perubahan yang terjadi pada nilai yang dideteksi. Bila v_1 dan v_2 adalah nilai sesungguhnya dan m_1 dan m_2 adalah nilai yang terukur sensor, maka sensitivitas adalah $sensitivitas = (m_2-m_1)/(v_2-v_1)$.

Error adalah perbedaan antara nilai yang diukur sensor dengan nilai sebenarnya. Bila v adalah nilai sesungguhnya dan m adalah nilai yang terukur sensor, maka $error = m-v$.

Akurasi adalah tingkat/persentase kepercayaan terhadap sensor. Bila v adalah nilai sesungguhnya dan error adalah nilai kesalahan pengukuran sensor, maka nilai akurasi didapat dengan rumus $akurasi = 1 - (error/v)$.

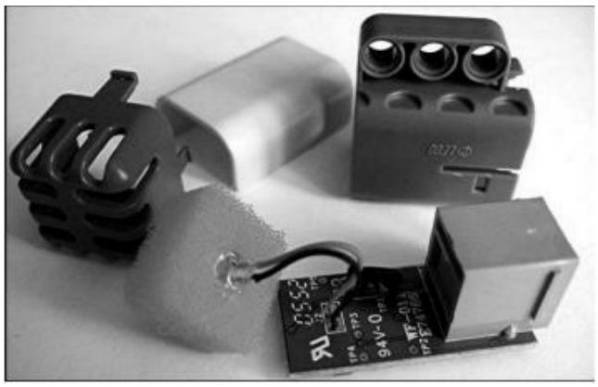
Presisi adalah sejauh apa konsistensi kesamaan nilai yang terukur sensor bila mengukur nilai sesungguhnya yang dibuat sama berkali-kali. Nilai presisi didapat dengan rumus $presisi = range/deviation$ dimana range adalah jangkauan nilai yang dapat diukur sensor dan deviation adalah standar deviasi random error berdasarkan data sama yang diukur berkali-kali.

2.2. Sensor Suara

Sensor suara adalah sensor yang dapat mendeteksi adanya suara. Frekuensi suara yang ditangkap oleh sensor suara LEGO Mindstorms NXT disesuaikan frekuensi pendengaran manusia yaitu antara 20 Hz – 20 Khz. Gambar berikut merupakan bentuk sensor suara LEGO Mindstorms NXT.

Gambar 1 Sensor Suara Bawaan LEGO

Bagian dalam sensor suara terdiri dari sebuah mikrofon yang terbungkus busa plastik, sebuah konektor kabel ke NXTBrick, serta sebuah PCB dimana rangkaian utama sensor terpasang. Berikut merupakan kelengkapan dan isi dari PCB (Printed Circuit Board) yang terdapat pada sensor suara.



Gambar 2 Sensor Suara yang Dibongkar

Sensor suara pada LEGO Mindstorms NXT dapat digunakan untuk mengukur intensitas suara pada suatu lingkungan. Sensor suara dapat mendeteksi suara dalam ukuran desibel (dB) dan adjusted desibel (dbA), mengetahui pola suara, serta perbedaan nada yang terdengar. Adjusted desibel adalah skala untuk mengukur intensitas suara dalam skala manusia

sedangkan desibel adalah skala untuk mengukur intensitas semua suara. Berikut adalah contoh suara dengan ukuran desibelnya.

90 dB	Suara yang keras
80 dB	Teriakan
70 dB	Berbicara
60 dB	Berbicara pada jarak yang cukup jauh
50 dB	Ruangan yang sunyi

Tabel 1Contoh Suara dan Ukuran db-nya

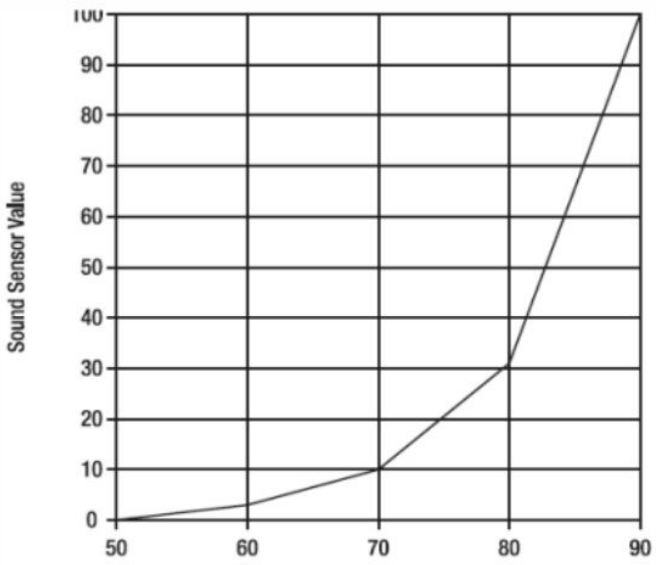
Kerja sensor suara dimulai ketika sensor menerima input suara dari lingkungan. Suara tersebut akan diterima oleh mikrofon yang ada pada sensor. Diafragma dalam mikrofon akan menangkap getaran suara dan ikut bergetar sesuai dengan frekuensi getaran yang ditangkap. Getaran tersebut akan menyebabkan magnet dalam mikrofon bergerak dengan frekuensi yang sama dengan frekuensi suara yang diterima. Magnet kemudian bergerak dalam kumparan yang selanjutnya menghasilkan aliran listrik dengan frekuensi yang sama pula. Aliran listrik inilah yang dikonversi menjadi sinyal yang sesuai agar dapat dibaca NXTBrick. NXTBrick selanjutnya mengkonversi sinyal tersebut dalam persentase antara 0 – 100%. Persentase tersebut dapat diartikan sebagai tingkat kekerasan suara yang diterima oleh sensor.

Berdasarkan nilai persentase tersebut, robot dapat memperkirakan kondisi lingkungan dimana input suara berasal. Tabel berikut menunjukkan pemahaman robot terhadap persentase kekuatan sinyal yang diperoleh.

4-5%	Ruang tamu yang hening
5-10%	Pembicaraan pada jarak tertentu
10-30%	Pembicaraan yang dekat dengan sensor atau musik dalam volume normal
30-100%	Teriakan atau musik yang dimainkan dalam volume tinggi

Tabel 2 Contoh Suara dan Ukuran Persentase LEGO-nya

Kembali menegaskan penjelasan sebelumnya, sensor suara dapat mendeteksi suara dalam ukuran desibel (dB) dan adjusted desibel (dbA). Hasil keluaran sensor akan ditampilkan dalam persentase. Untuk diperlukan suatu pemetaan dari nilai suara yang diperoleh ke dalam ukuran persentase. Grafik di bawah ini menunjukkan model pemetaan yang digunakan oleh NXTBrick.



Gambar 3 Relasi Antara Desibel dan Persentase LEGO

3 METODE RISET

3.1 Pengembangan

1. **Desain Kaki Robot**. Konstruksi mekanisme berjalan dengan dua kaki.
2. **Desain Tubuh dan Kepala**. NXT Brick sebagai tubuh dan sensor ultrasonik sebagai kepalanya.
3. **Tangan**. Mekanisme tangan dimana di ujung tangan kiri terdapat sensor sentuh dan di ujung tangan kanan terdapat sensor suara.

3.2 Programming

Menggunakan LeJOS

3.3 Teknik Riset

Setelah Robot selesai dikonstruksi, maka akan dibuat program/perangkat lunaknya untuk meneliti dan memahami kinerja sensor suara menggunakan LeJOS.

Setelah peneliti memahami kapabilitas sensor barulah peneliti membuat program untuk skenario yang disebutkan di bagian sebelumnya.

Kemudian peneliti mencoba memikirkan suatu teknik untuk mengimprovisasi kinerja pengenalan percakapan yang diindera oleh sensor suara yang konon katanya masih relatif buruk.

Untuk mengukur sensitivitas, error, akurasi dan presisi akan digunakan volume suara dari komputer dengan bantuan perangkat lunak audio, sehingga kerasnya suara bisa dikendalikan secara tepat dan terukur.

4. HASIL EXPERIMEN

5. KESIMPULAN DAN SARAN

Beberapa kesimpulan yang dapat kami tarik dari proyek kami ini adalah:

Beberapa saran dari kami:

TUGAS UTS (KELOMPOK)

KELAS ROBOTIKA SEMESTER GENAP 2011-2012

Dosen: Dr. Eng. Wisnu Jatmiko

Asisten: M. Sakti Alvissalim, S. Kom

DR. FRANKENSTEIN'S EAGLE

GAMBAR 1 RUN, RABBIT, RUN!!!

Tujuan :

Tugas ini bertujuan untuk memperdalam tingkat pemahaman mahasiswa tentang arsitektur ROS dan beberapa tools yang berguna di dalamnya.

Overview:

Dikisahkan bahwa Dr. Frankenstein ingin menciptakan sebuah monster baru yang menyerupai seekor elang. Dr. Frankenstein ingin menciptakan monster baru tersebut agar monster dia tidak perlu repot-repot lagi berburu binatang favoritnya, yakni kelinci. Pada saat berburu kelinci, dokter gila ini melihat bahwa elang dapat dengan mudah menangkap kelinci karena kemampuannya untuk mengamati mangsa sambil terbang. Elang juga dapat bermanuver dengan bebas dan dapat bergerak dengan cepat sehingga kelinci tidak punya waktu untuk menghindar.

GAMBAR 2 MONSTER CIPTAAN DR. FRANKENSTEIN (AR.DRONE)

Terinspirasi dari pengamatan sederhana tersebut, Dr. Frankenstein berusaha mengumpulkan sebongkah besi yang kemudian dia rakit dan letakkan di atas atap rumah saat petir menyambar. Petir akhirnya menyambar dan seketika itu sebongkah besi tersebut bisa terbang. Namun, sayangnya monster baru ciptaannya tersebut belum begitu cerdas, sama seperti monster pertama ciptaannya yang sangat primitif. Monster tersebut diberi nama AR.Drone. AR.Drone saat ini hanya bisa terbang di tempat dan mengerti perintah-perintah untuk melakukan gerakan sederhana seperti maju, mundur, dsb. Dokter yang sangat ambisius ini berusaha setengah mati untuk membuat AR.Drone lebih pintar sehingga bisa menangkapkan kelinci untuknya. Berbagai cara telah dia coba untuk membuat AR.Drone lebih cerdas, tetapi tidak yang membuahkan hasil. Di tengah keputusasaan, dia meminta bantuan teman-teman dekatnya, para peserta kuliah Robotika Semester Genap 2011/2012, untuk membantunya membuat AR.Drone lebih cerdas. Kalian diminta untuk mengajari AR.Drone agar bisa mengikuti pergerakan kelinci yang berlari di bawahnya. Sebagai permulaan Dr. Frankeinstein telah setuju untuk menggunakan sebuah benda berwarna sebagai representasi kelinci.

Tools:

Berikut adalah tools yang bisa digunakan:

- Robot Operating System (ROS)

Diasumsikan setiap peserta kuliah telah berhasil menginstal ROS.

- ardrone_brown

Node ini digunakan untuk mengontrol dan menerima data dari ARDrone. Diasumsikan setiap peserta kuliah telah berhasil menginstal ardrone_brown.

- cmvision

Digunakan untuk mendeteksi blob.

[http://mirror.umd.edu/roswiki/cmvision.html]

Instalasi cmvision:

- cd <path_to_your_ros_workspace>

- svn checkout https://code.ros.org/svn/wg-ros-pkg/branches/trunk_cturtle/vision/cmvision

Setelah tahap ini, source code dari cmvision akan berada dalam folder cmvision. Agar bisa dijalankan, perlu dilakukan kompilasi terlebih dahulu. Jalankan perintah berikut ini:

- cd cmvision

- rosmake --rosdep-install

Jika instalasi gagal, kemungkinan ada dependency yang belum terpenuhi. Salah satu dependency yang mungkin belum terpenuhi adalah pixmap. Jalankan perintah berikut ini untuk menginstal pixmap:

- sudo apt-get install gtk2-engines-pixbuf

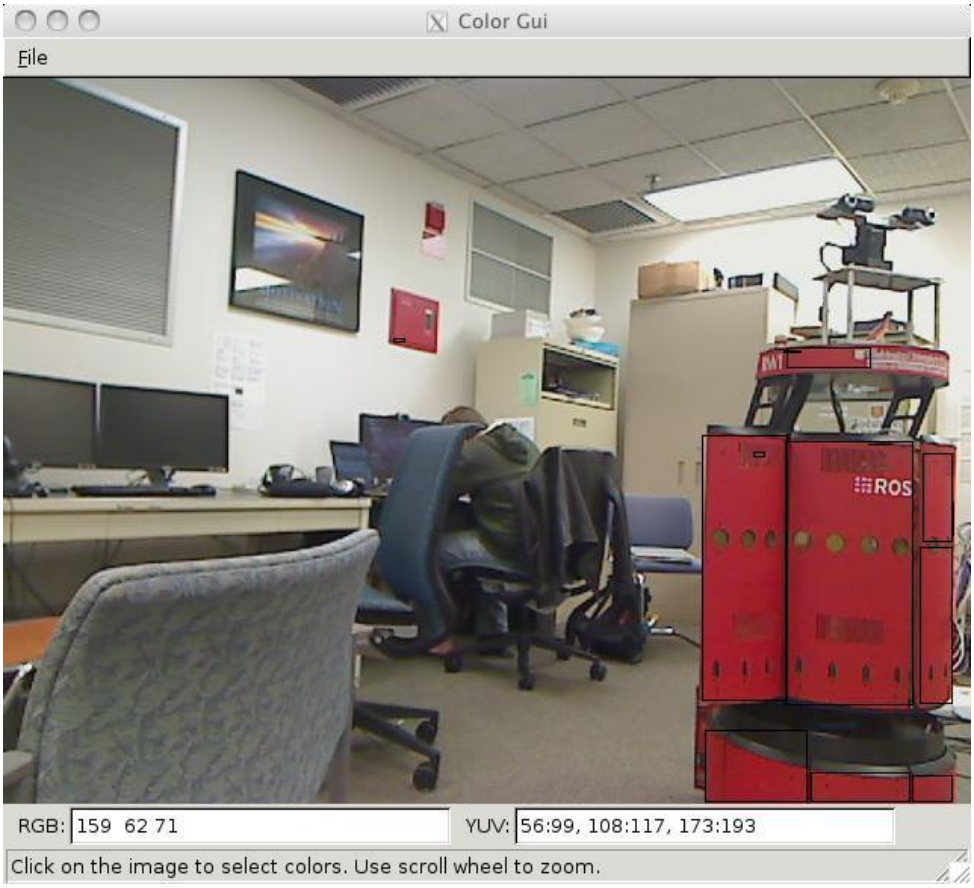
Jalankan ulang perintah rosmake --rosdep-install

Penggunaan cmvision:

Terdapat dua mode dalam menjalankan cmvision. Mode pertama adalah mode *colorgui*. Pastikan ardrone_brown telah dijalankan sebelumnya. Untuk menjalankan colorgui, jalankan perintah:

- rosrun cmvision colorgui image:=/ardrone/image_raw

Akan muncul GUI seperti pada Gambar 3. Kita bisa memilih warna yang ingin kita deteksi dengan menggunakan GUI yang telah disediakan dengan meng-klik area yang ingin dideteksi.



GAMBAR 3 CMVISION COLOR GUI

Area yang dideteksi ditunjukkan oleh kotak-kotak. Jika hasil pendeteksian dinilai sudah cukup baik, kita bisa mencatat nilai RGB dan YUV yang ditampilkan pada GUI.

Pastikan bahwa nilai RGB dan YUV telah dicatat dan tutup program dengan menekan Ctrl + C pada keyboard.

Kemudian buat text-file baru bernama color_spec.txt pada direktori cmvision.

- gedit color_spec.txt Tulis nilai RGB dan YUV dengan format seperti di bawah ini.

Tulis nilai RGB di bawah [colors] dan nilai batas YUV di bawah [thresholds].

```
[colors] (159, 62, 71) 0.000000 7 ResearchRobotRed

[thresholds] (56:99, 108:117, 173:193)
```

Simpan file tersebut. Untuk menjalankan program cmvision dan mulai mempublish topik blob, download file launcher yang telah diupload di scele (*cmvision_ardrone.launch*) dan copy ke direktori cmvision. Kemudian jalankan cmvision dengan perintah berikut:

- roslaunch cmvision cmvision_ardrone.launch

Informasi blob yang dideteksi dipublish ke topik /blobs. Untuk mendapatkan informasi lebih detail mengenai topik tersebut dan apa isinya jalankan perintah berikut ini.

- rostopic echo /blobs

ardrone_brown:

Jalankan node ini dengan perintah:

- rosrun ardrone_brown ardrone_driver

Secara default, ardrone_brown akan mempublish image dari kamera depan ARDrone. Untuk mendapatkan gambar dari kamera bawah ardrone jalankan perintah berikut:

- rosservice call /ardrone/togglecam

Ketentuan:

Pada worksheet ini diharapkan peserta kuliah dapat memahami penggunaan node cmvision untuk pendeteksian obyek sederhana melalui kamera ARDrone.

- Gunakanlah node cmvision untuk melakukan pendeteksian object satu warna sederhana

- Buatlah sebuah node di ROS yang bisa mensubscribe topik /blobs dan memanfaatkan informasi tersebut untuk membuat ARDrone mengikuti pergerakan benda yang dideteksi. Gunakan node ardrone_brown untuk mengirimkan perintah pergerakan dan menerima data dari ARDrone.

- Object yang digunakan untuk merepresentasikan kelinci bisa ditentukan secara bebas.

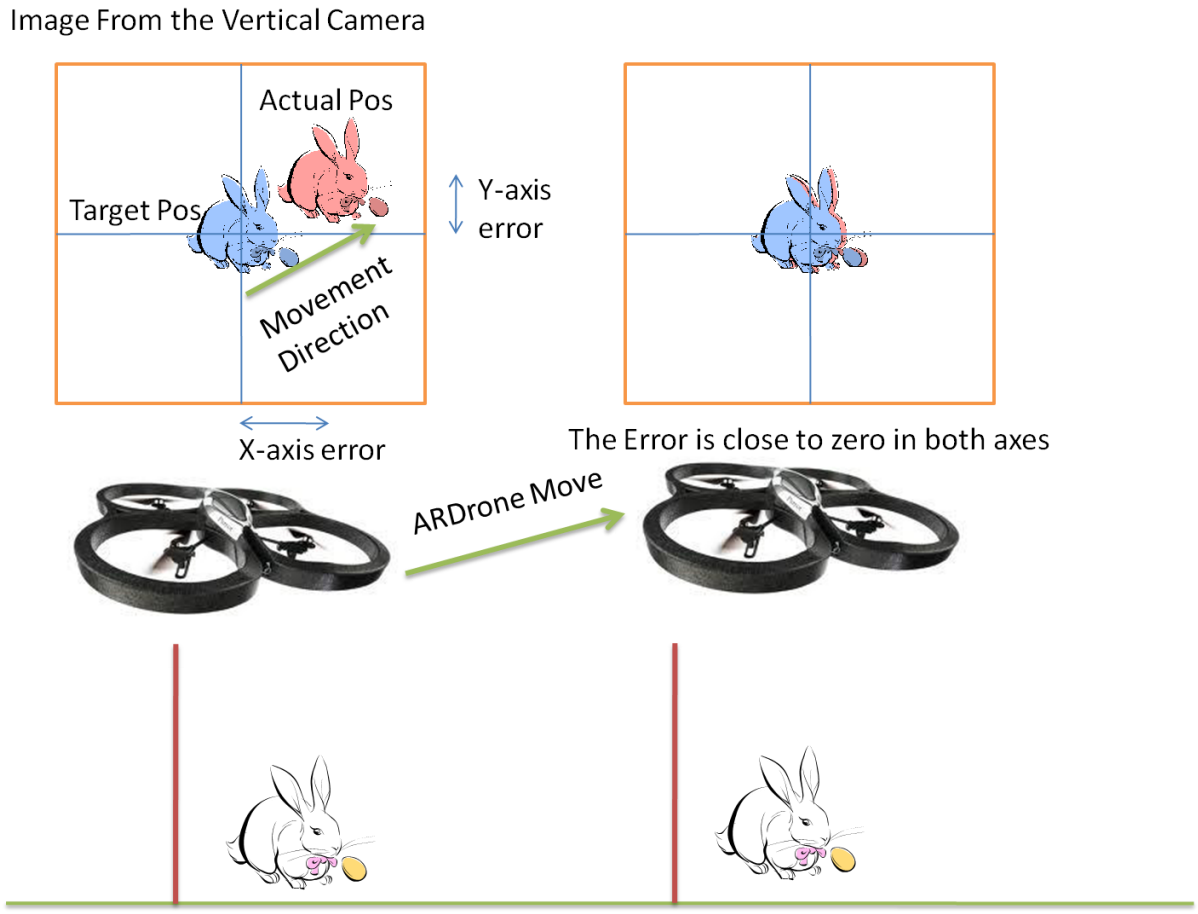
- Object digerakkan secara manual maupun otomatis. Pergerakan object secara otomatis akan mendapatkan nilai bonus.

- Diasumsikan ketinggian AR.Drone dijaga agar tetap sama sehingga hanya pergerakan pada sumbu X dan Y yang perlu dilakukan.

- Target yang ingin dicapai ARDrone adalah untuk selalu mengikuti object sehingga object selalu berada di tengah kamera bawah ARDrone.

Gambar 4 merupakan ilustrasi misi yang ingin dicapai. Setelah take-off, ARDrone mendeteksi keberadaan kelinci melalui kamera bawahnya. Pada gambar kamera bawah, gambar kelinci terletak tidak di tengah gambar, ini artinya ARDrone tidak berada tepat di atas kelinci. Perbedaan posisi kelinci terhadap titik tengah gambar disebut sebagai error. Tujuan dari

ARDrone adalah selalu menjadikan nilai error ini mendekati 0 pada setiap saat dengan cara melakukan perpindahan agar posisi kelinci bisa berada tepat di tengah gambar.



GAMBAR 4 ILUSTRASI TARGET YANG INGIN DICAPAI

Variasi Tugas:

Terdapat 5 variasi spesifikasi tambahan yang bisa diimplementasikan. Setiap kelompok harap memilih paling tidak satu. Buatlah sebuah node yang terpisah dari node program utama untuk mengimplementasikan salah satu fungsionalitas di bawah ini:

a. Menunjukkan perkiraan posisi ARDrone

   Buatlah sebuah node yang berfungsi menampilkan perkiraan posisi ARDrone menggunakan data kecepatan dari Navdata. Perhatikan bahwa posisi dapat didapatkan dari kecepetan dengan mengintegrasikan kecepatan terhadap waktu.

   Hint: Integrasikan vx, vy, dan vz dari navdata untuk mendapatkan posisi relatif ARDrone, kemudian transformasikan posisi relatif ke posisi dalam koordinat absolut (Ground Frame).

Publish topik bernama "/drone_pos" yang berisi posisi absoult (**geometry_msgs/Pose**) ARDrone dan tampilkan menggunakan rx_graph.

b. Menunjukkan perkiraan posisi relatif kelinci terhadap AR.Drone

   Buatlah sebuah node yang berfungsi menampilkan perkiraan posisi relatif kelinci terhadap ARDrone . Posisi ARDrone menjadi titik (0,0) dalam sistem koordinat tersebut dan perbedaan posisi kelinci dengan ARDrone dikeluarkan setiap waktunya.

   Publish topik bernama "/rabbit_pos_relative" (**geometry_msgs/Pose**) yang berisi posisi relatif kelinci.

c. Memperkirakan kecepatan dan arah pergerakan kelinci.

   Perkirakan kecepatan kelinci relatif terhadap ARDrone dengan menghitung perbedaan posisi kelinci dalam jangka waktu tertentu.

   Publish topik bernama "/rabbit_vector" **(geometry_msgs/Vector3)** berisi kecepatan dalam sumbu x, y, dan z.

d. Menunjukkan kecepatan absolut kelinci

   Hitung kecepatan absolut kelinci dengan menghitung kecepatan relatif kelinci terhadap ARDrone dan kecepatan Absolut ARDRone.

   Publish topik bernama "/rabbit_pos_absolute" **(geometry_msgs/Pose)** berisi kecepatan dalam sumbu x, y, dan z.

e. Menunjukkan posisi absolut dari kelinci

   Kombinasikan informasi posisi ARDrone dengan posisi relatif kelinci untuk mendapatkan posisi Absolut kelinci.

   Publish topik bernama "/rabbit_pos_absolute" **(geometry_msgs/Pose)** berisi posisi dalam sumbu x, y, dan z.

Beberapa fitur spesifikasi diatas hanya dapat dipenuhi jika spesifikasi yang lain berhasil dipenuhi. Berikut adalah dependency list dari masing-masing spesifikasi.

a : none

b: none

c: b

d: c <- b

e: d <- c <- b

Oleh karena ketergantungan di atas, kelompok-kelompok yang saling berhubungan diperbolehkan bekerja sama untuk implementasi tugas variasi.

Pengumpulan Tugas:

Tugas harus dikumpulkan dalam bentuk sebuah file .zip berisi:

1. *Source code* terdiri dari satu folder node yang dikembangkan oleh kelompok beserta isinya
2. Dokumentasi penggunaan program berisi dalam format .pdf berisi:
    a. Cara instalasi node
    b. Cara menjalankan program
    c. Penjelasan cara kerja dan teknik yang digunakan
    d. Pembagian kerja dalam kelompok

    Banyak halaman sekitar 15.
3. Video dokumentasi hasil pergerakan AR.Drone yang mengikuti ketentuan sebagai berikut:
    a. Disimpan dalam format .avi, .mpg, atau .mp4
    b. Durasi minimal 2 menit dan maksimal 5 menit.
    c. Dimulai dari rekaman proses menjalankan program dan tampilan user interface (jika ada), kemudian dilanjutkan dengan rekaman pergerakan ARDrone saat mengikuti object.
    d. Usahakan kamera tidak melakukan banyak pergerakan dan tidak banyak bergoyang-goyang.

Tugas harus dikumpulkan melalui slot yang disediakan di Scele sebelum tanggal 13 April 2012.

Format penamaan:

Kelompok[Nomor Kelompok]_[Group's Wireless Network Name].zip

Akan diadakan sesi presentasi pada tanggal 13 dan 14 April 2012. Jadwal presentasi akan diumumkan kemudian.

Penilaian:

- **Program** : 40 %
  Penilain program meliputi fungsionalitas program, kerapian dan keterbacaan. Fungsionalitas dinilai berdasarkan kemampuan mendeteksi object, kemudian kemampuan untuk mengikuti object.
- **Dokumentasi** : 20 %
  Dokumentasi dinilai berdasarkan kelengkapan poin-poin yang telah disebutkan.
- **Video** : 20 %
  Dinilai berdasarkan kejelasan video dalam menyampaikan informasi
- **Presentasi** : 20%

Dinilai berdasarkan penampilan saat presentasi.

Selamat membantu Dr. Frankenstein untuk membuat Elang ciptaannya lebih cerdas. Pastikan kalian berhasil atau kalian akan menjadi bahan percobaan Dr. Frankenstein selanjutnya.... :D

Good Luck. :D

NB:

Pengerjaan bisa dimulai 03 April 2012.

ARDrone bisa mulai dipinjam dan tempat eksperimen yang bisa digunakan adalah lorong gedung A Fasilkom UI dari depan Lab 1231 hingga tangga.

Lorong tersebut boleh dipakai setiap hari di atas pukul 18.00 hingga 13 April 2012.

TUGAS PENGGANTI

UJIAN AKHIR SEMESTER (UAS)

ODOR SOURCE LOCALIZATION – NEXT GENERATION

(OSL-NG)

KELAS ROBOTIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA

2012/2013

I. Pendahuluan

Pada tahun 2013 ini terjadi berbagai fenomena alam yang meresahkan masyarakat sekiar. Salah satunya adalah **global warming**, dimana terjadi kenaikan suhu di permukaan bumi. Gejala ini dipicu oleh berbagai sebab, diantaranya adalah, polusi dari mesin-mesin industri, banyaknya zat-zat yang mencemari atmosfer, dan yang paling parah adalah berkurangnya tanaman hijau di bumi khususnya hutan. Setelah disinyalir, ternyata selama sepuluh terakhir ini terjadi berbagai kasus penebangan dan pembakaran hutan untuk dijadikan lahan pertanian maupun pemukiman. Hal ini jelas-jelas merupakan permasalahan yang serius dan harus segera diatasi.

Anda sebagai calon teknokrat bangsa Indonesia yang cinta terhadap bumi dan lingkungan Indonesia diberikan sebuah tugas mulia. Tugas mulia tersebut adalah mengembangkan sistem robot swarm untuk mencari lokasi sumber asap (odor source localizaion) yang ada di wilayah hutan Indonesia. Dengan menemukan sumber asap, maka juga ditemukan sumber kebakaran untuk segera ditangani. Seperti yang kita ketahui bahwa untuk menjelajahi kawasan hutan, diperlukan suatu alat transportasi yang cepat. Hal ini jelas saja tidak dapat dijangkau oleh robot darat maupun robot air. Oleh sebab itu anda diminta menggunakan koloni *Unmanned Aerial Vehicle* (UAV) yang dalam hal ini adalah *quadcopter* dalam menyelesaikan misi terebut. Selain itu, pengalaman meggunakan quadcopter yang anda miliki akan sangat berguna untuk menyelesaikan misi ini.

Meskipun demikian, sebelum diimplementasikan ke dalam sistem koloni *quadcopter* yang sebenarnya, anda diminta untuk mengimplementasikannya dalam suatu simulasi terlebih

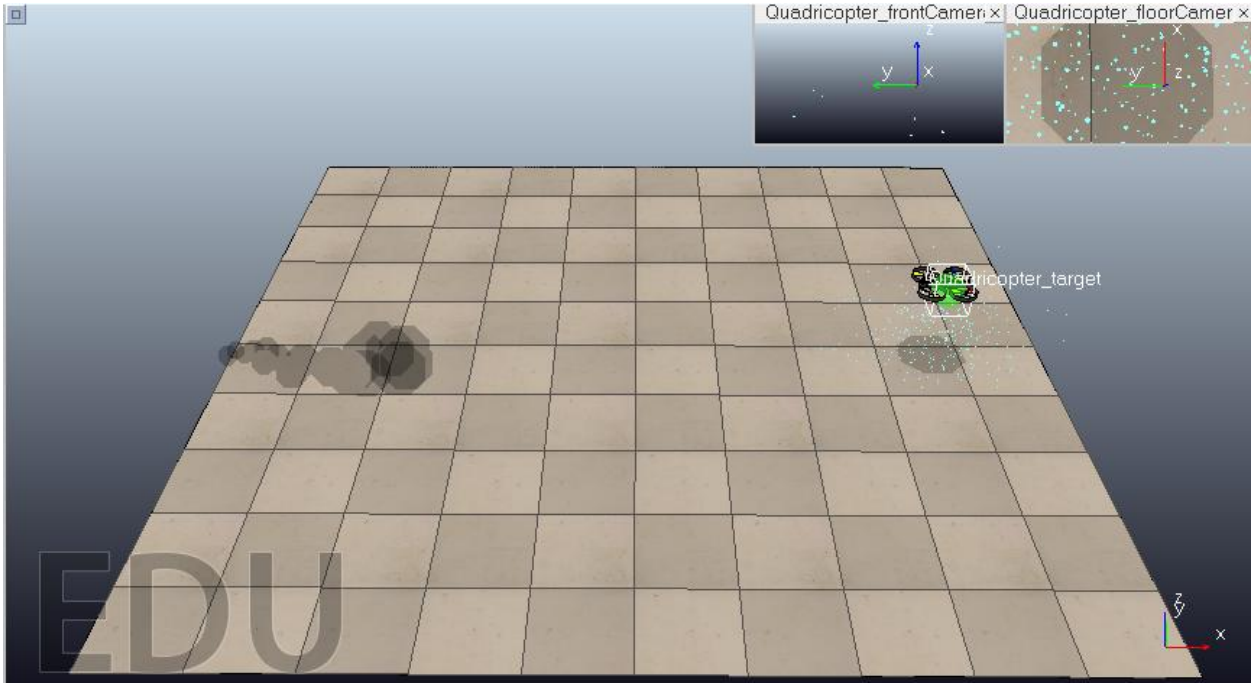
dahulu. Hal ini dimaksudkan untuk mengurangi resiko kecelakaan dan kerusakan dalam implementasi. Simulator yang akan digunakan di sini adalah simulator V-Rep. Dengan demikian, setelah algoritma robot swarm yang dikembangkan untuk misi *odor source localization* ini berhasil diterapkan, maka selanjutnya dapat diimplementasikan ke dalam sistem yang sebenarnya.

II. Deskripsi Tugas

Dalam tugas ini anda diminta untuk mengimplementasikan algoritma robot swarm untuk mencari keberadaan sumber asap. Algoritma yang dikembangkan harus dapat memenuhi 2 kondisi utama, yaitu :

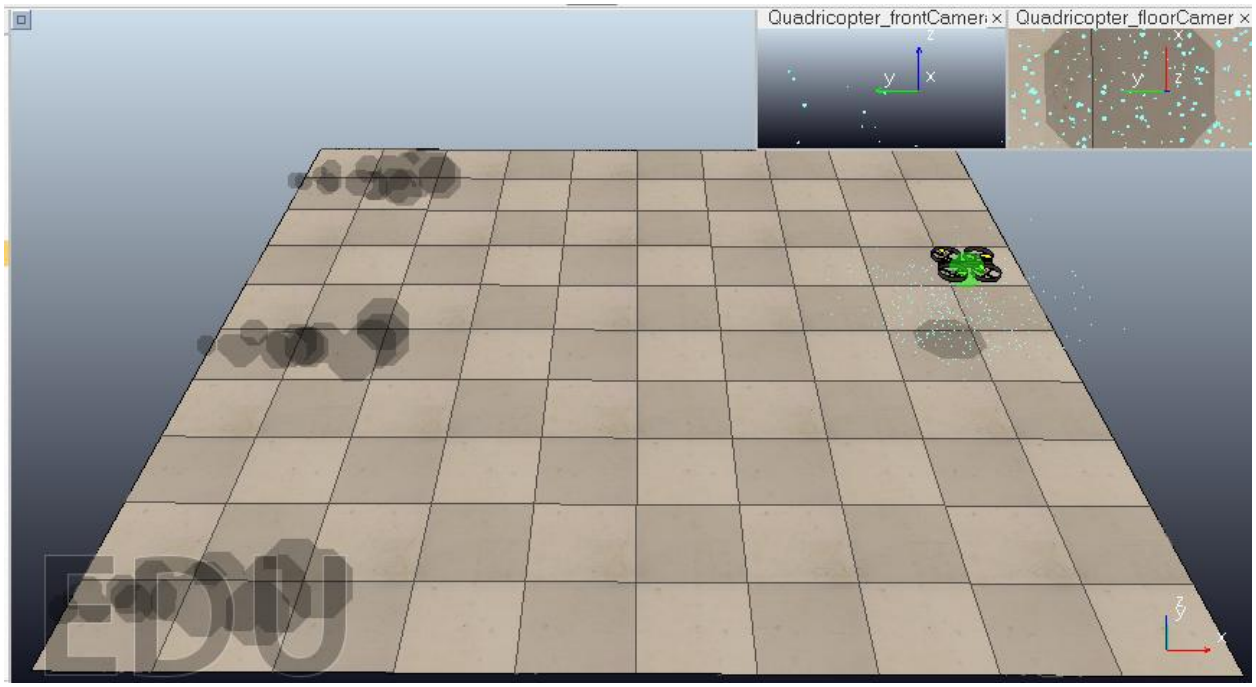
a. Kondisi dimana hanya ada satu sumber asap

Ilustrasi kondisi satu sumber asap



b. Kondisi dimana ada lebih dari 1 sumber asap (3 sumber)

Ilustrasi kondisi 3 sumber asap



Untuk memodelkan konsentrasi sumber asap di setiap lokasi lapangan, digunakan metode distribusi Gaussian. DIstribusi Normal, dimana :

$$C = 1/\ sqrt(2\ .\pi\ .stdev)\ \ eks\ \ (\ -\ d^2\ /\ 2(stdev^2))$$

Dimana C= konsentrasi asap di suatu titik. Stdev adalah standart deviasi distribusi Gaussian yang digunakan (dalam tugas ini stdev=0.1), d= jarak antara sumber posisi robot dan sumber asap (**data posisi sumber asap tidak boleh digunakan kecuali untuk menghitung konsentrasi**). dan eks(a) = e^a , e = 2,718. Contoh penghitungan konsentrasi sumber asap (satu sumber) pada kode V-Rep :

```
posX = simGetFloatSignal('odorPosX')
posY = simGetFloatSignal('odorPosY')
stdev = 0.1
dx = ((posX-pos[1])^2) + ((posY-pos[2])^2)
con = 1 / (stdev * 2* 3.14) * (2.718 ^ (-(dx / (2*stdev*stdev))))
simAddStatusbarMessage('Concentration')
simAddStatusbarMessage(con)
```

Dalam kasus multi sumber, akan dihitung Ci (konsentrasi dari masing-masing sumber). Lalu konsentrasi total Ct = C1+C2+ … Cn (n = banyak sumber).

Rincian Tugas :

1. Menerapkan algoritma swarm robot untuk mencari sumber asap untuk kedua kasus di atas. Algoritma yang digunakan bebas, baik *Particle Swarm Optimization* (PSO) standar maupun variasinya. **Sebagai bantuan akan diberikan template 'scene' untuk kedua skenario**
2. Jumlah robot yang digunakan untuk kasus single source adalah 5 robot, dan untuk kaus multiple source adalah 10 robot.
3. **Data posisi sumber asap tidak boleh digunakan kecuali untuk menghitung konsentrasi**
4. Skenario yang digunakan untuk tiap kasus minimal ada 2. Skenario pertama adalah semua robot start dari posisi yang sama (berdekatan). Skenario kedua robot start dari posisi berjauhan (random). Akan tetapi robot harus start dari posisi dimana konsentrasi asap = 0.
5. Analisis minimal mencakup tentang performa algoritma yang dapat diukur dengan waktu yang dibutuhkan untuk menemukan lokasi sumber asap.
6. Bonus diberikan jika ditambahkan scenario berikut (boleh salah satu atau keduanya):
   a. Penambahan obstacle pada lapangan
   b. Pembandingan performa penggunaan jumlah robot dalam masing-masing scenario

III.Pengumpulan

Berkas-berkas yang harus dikumpulan adalah sebagai berikut :

1. Laporan akhir percobaan, format laporan sama dengan UTS
2. Presentasi
3. Video masing-masing skenario
4. Source code

Deadline tugas ini adalah 20 Juni 2013

IV.Proporsi Penilaian

Proporsi Penilaian :

1. Eksperimen 70 % (35 % tiap skenario)
2. Laporan 10 %
3. Presentasi 5 %
4. Video demo 5%
5. Source code 10%
6. Bonus maksimal 10%

UJIAN TENGAH SEMESTER

ROBOTIKA

SEMESTER GENAP TAHUN AJARAN 2014-2015

TEMA : FIRE FIGHTING ROBOT

SIFAT : TUGAS KELOMPOK

Oleh:

Wisnu Jatmiko

Muhammad Anwar Ma'sum

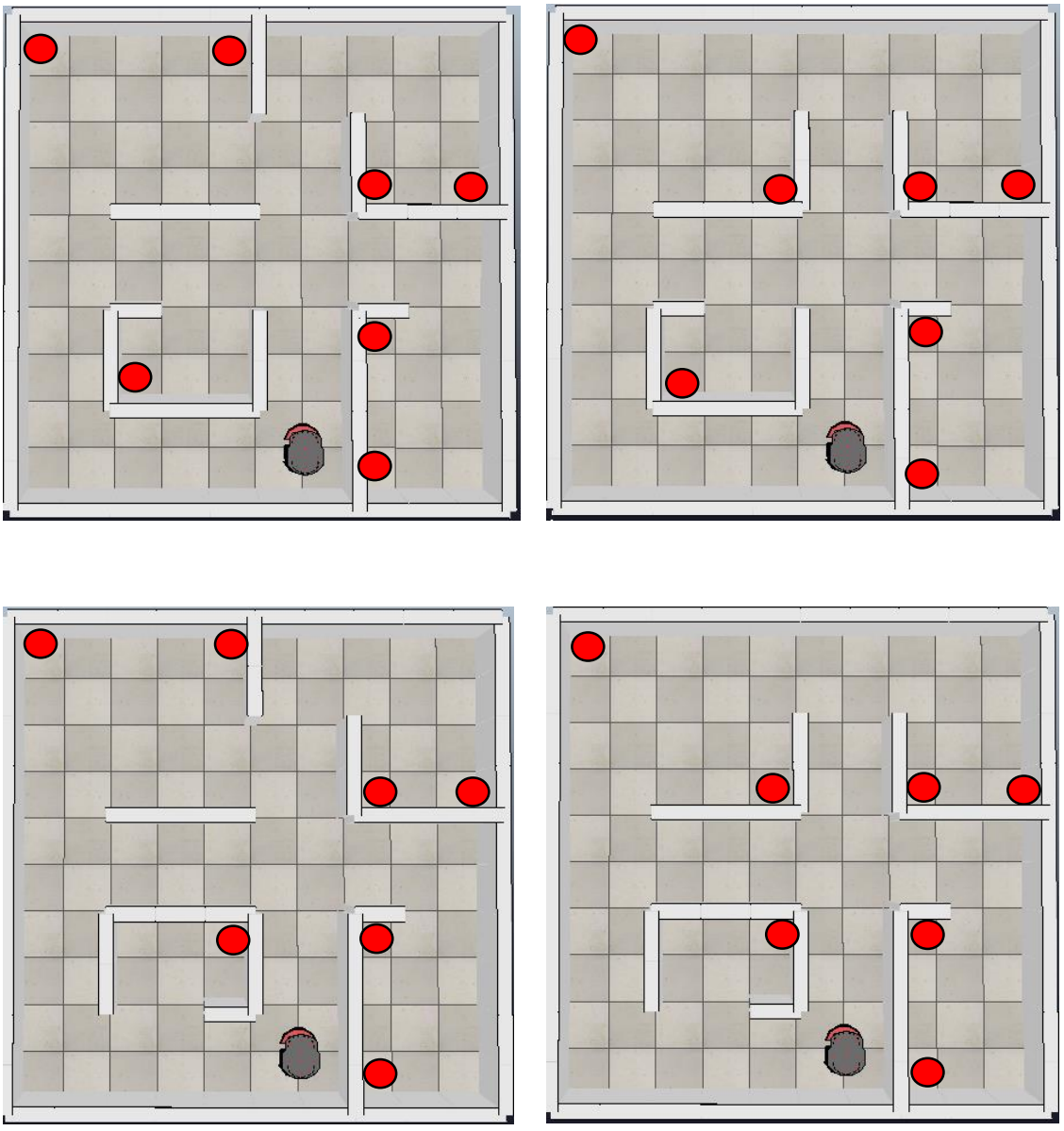
FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA

2015

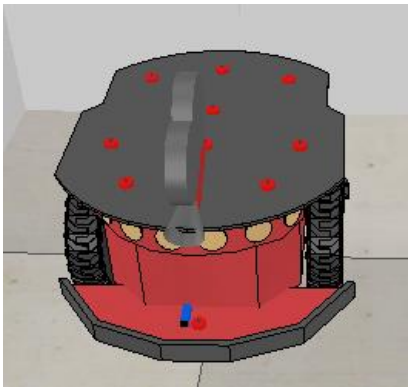
I. Pendahuluan

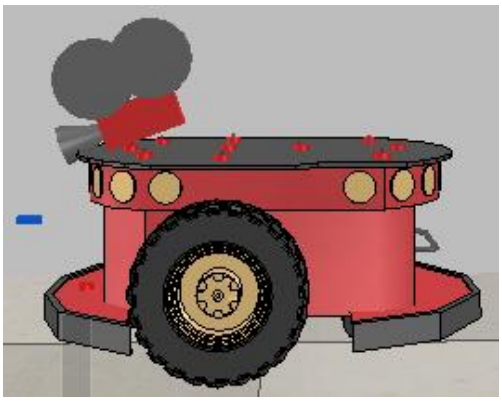
Seiring dengan perkembangan ilmu pengetahuan dan teknologi di bidang robotika, permintaan dan kebutuhan akan robot di masyarakat juga semakin meningkat. Salah satu jenis robot yang populer di masyarakat saat ini adalah robot pemadam api. Robot ini diperlukan karena untuk menjalankan tugas yang berbahaya dan beresiko tinggi, yakni mencari dan memadamkan api. Robot ini bekerja secara autonomous, yang mana user cukup menekan tombol start dari titik start yang diberikan. Selanjutkan robot akan mencari api di suatu ruangan (maze) yang diberikan. Sebagai generasi muda yang memiliki minat dan bakat di bidang robotika, anda diminta untuk mengimplementasikan robot pemadam api guna memebuhi permintaan masyarakat tersebut.



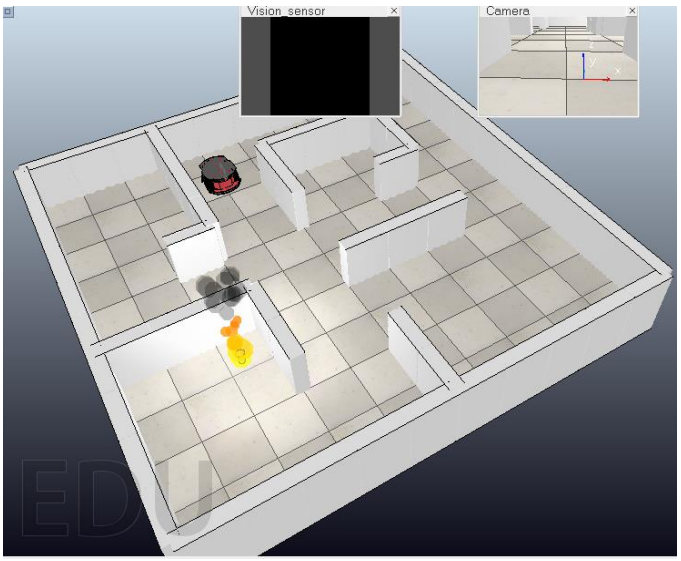
II. Soal

Diberikan suatu rumah yang mana terdapat api di dalamnya. Karena rumah ini tergolong rumah yang moderen dan canggih, maka konfigurasi ruangannya pun bisa berubah-ubah. Totalnya, ada empat kemungkinan konfigurasi. Keempat konfigurasi rumah tersebut dapat dilihat pada gambardi atas. Titik-titik berwarna merah adalah posisi-posisi yang mungkin terdapat api (Hanya ada sat api). Pada tugas kali ini, robot yang digunakan adalah robot pioneer, seperti yang ditampilkan gambar berikut.





Tugas robot pioneer adalah menjelajahi ruangan (maze), menemukan api dan mematikannya. Pada template soal, sudah disediakan map dan robot pioneer yang dilengkapi sensor ultrasonik, sensor kamera dan vision, serta denah ruangan, serta denah ruangan. Berikut tampilan salah satu map. Untuk mengubah dari satu map ke map lain dapat dilakukan dengan modifikasi lokasi (Wall 49,Wall50, Wall57,Wall58,Wall59).



Adapun tugas yang harus kalian selesaikan adalah sebagai berikut :

1. Tambahkan (pasang) aktuator berupa air jet atau sejenisnya pada robot untuk keperluan memamdamkan api saat robot sudah menemukan api. Berikut adalah gambar air jet

2. Carilah keterbatasan sensor-sensor yang digunakan pada robot.
3. Rancang dan implementasikan algoritma untuk menemukan api untuk keempat map yang ada. Sebagai catatan, algoritma yang dibuat hanya satu, akan tetapi mampu menangani keempat jenis map (bukan membuat satu algoritma untuk setiap map)
4. Lakukan eksperimen dan analisis, minimal ada skenario-skenario berikut :
   a. Mencari keterbatasan sensor-sensor yang digunakan pada robot.
      Catat dalam tabel berikut :

No	Jenis Sensor	Keterbatsan	Kondisi yang menyebabkan sensor tidak bekerja sesuai dengan keinginan
1			

   b. Buatlah eksperimen dengan variasi kecepatan motor robot, lalu lakukan analisis terhadap tingkat keberhasilan mencari api dan waktu menemukan api. Rangkumlah dalam tabel-tabel berikut. Setiap sel pada tabel-tabel berikut adalah rata-rata dari 5 kali perulangan eksperimen.

Tabel pengaruh kecepatan robot terhadap keberhasilan

Kecepatan roda	Rata-rata tingkat keberhasilan (%)				Kondisi yang menyebabkan gagal
	Map1	Map2	Map3	Map4	
1					
2					
3					

Tabel pengaruh kecepatan robot terhadap waktu pencarian

Kecepatan roda	Rata-rata lama waktu pencarian(%)			
	Map1	Map2	Map3	Map4
1				
2				
3				

5. Buatlah plot kurva kecepatan roda robot selama simulasi berjalan. Berikan analisis dan penjelasan tentang kurva tersebut
6. Simpulkan hasil eksperimen anda
7. Buatlah laporan dokumentasi pengerjaan tugas tersebut.

III. Penilaian

Berikut ada;ah komponen penilaian tugas UTS kelompok :

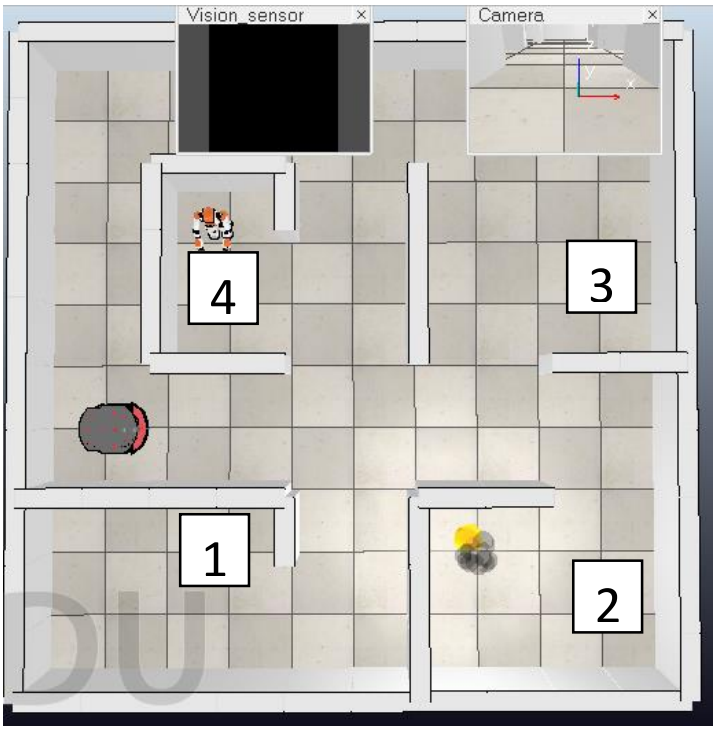
- 50% Program dapat berjalan dengan sukses sesuai permintaan soal
- 25% Presentasi dan demo
- 25% Laporan (baik konten maupun penulisan)

Bonus (max 20%) :

Jika berhasil menambahkan fitur lapor ke pemilik rumah. Setelah memadamkan api, maka robot akan mencari dan mendatangi pemilik rumah untuk melapor bahwa api sudah dipadamkan. Pemilik rumah dimodelkan dengan robot NAO. Robot pemadam api dan NAO harus berkomunikasi menggunakan trasreceiver. Berikut adalah posiss si pemilih rumah :

- Jika api di ruang 1, maka pemilik rumah di ruamg 3
- Jika api di ruang 2, maka pemilik rumah di ruamg 4
- Jika api di ruang 3, maka pemilik rumah di ruamg 1
- Jika api di ruang 4, maka pemilik rumah di ruamg 2

Gambar dibawah merupakan contoh scene yang terdapat pemilik rumah. Sebagai catatan, untuk melapor, robot pemadam api harus pergi ke ruangan tempat pemilik rumah berada.



IV. Pengumpulan

Kumpulkan pengerjaan tugas kelompok anda dengan format “UTS_kelompok[nomor].zip” yang berisi file .ttt, laporan/dokumentasi, dan presentasi. Keterlambatan pengumpulan akan mendapat pengurangan 5 poin setiap hari.

V. Presentasi, Demo dan Tanya Jawab

Setiap kelompok akan mendapatkan waktu total untuk presentasi, demo, dan tanya jawab minimal 30 menit. Adapun alur demonstrasi adalah seperti berikut :

1. Peserta memilih jenis map dan lokasi api, lalu simulator dijalankan
2. Penguji memilih jenis map dan lokasi api secara acak, lalu simulator dijalankan. Pengujian ke-2 dilakukan dengan mengganti posisi Wall tertentu, tidak berganti / load file baru, untuk membuktikan robustness algoritma yang sudah diimplementasi.

UJIAN AKHIR SEMESTER

ROBOTIKA

SEMESTER GENAP TAHUN AJARAN 2014-2015

TEMA : FIRE FIGHTING ROBOT NEXT GENERATION

SIFAT : TUGAS KELOMPOK

Oleh:

Wisnu Jatmiko

Muhammad Anwar Ma'sum

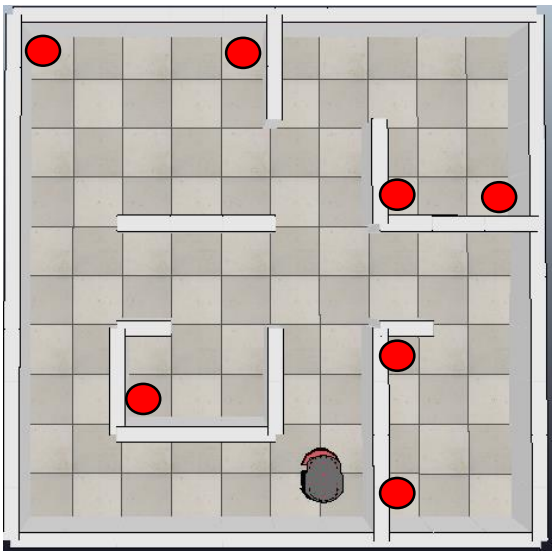
FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA

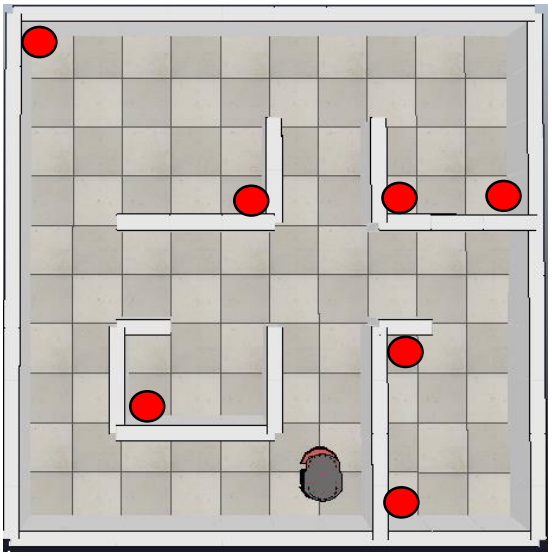
2015

I. Pendahuluan

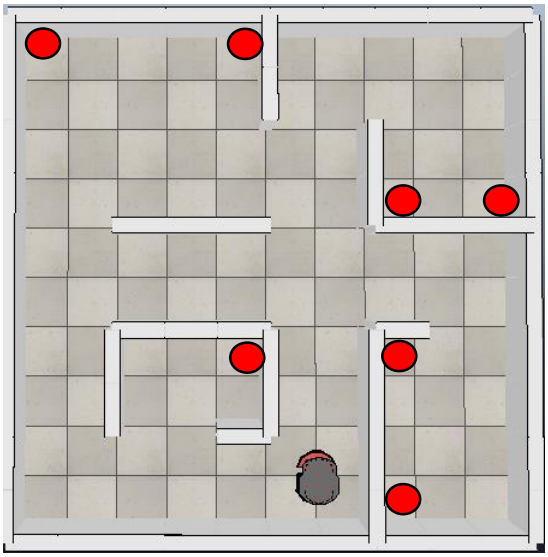
Seiring dengan perkembangan ilmu pengetahuan dan teknologi di bidang robotika, permintaan dan kebutuhan akan robot di masyarakat juga semakin meningkat. Salah satu jenis robot yang populer di masyarakat saat ini adalah robot pemadam api. Robot ini diperlukan karena untuk menjalankan tugas yang berbahaya dan beresiko tinggi, yakni mencari dan memadamkan api. Robot ini bekerja secara autonomous, yang mana user cukup menekan tombol start dari titik start yang diberikan. Selanjutkan robot akan mencari api di suatu ruangan (maze) yang diberikan. Sebagai generasi muda yang memiliki minat dan bakat di bidang robotika, anda diminta untuk mengimplementasikan robot pemadam api guna memebuhi permintaan masyarakat tersebut.



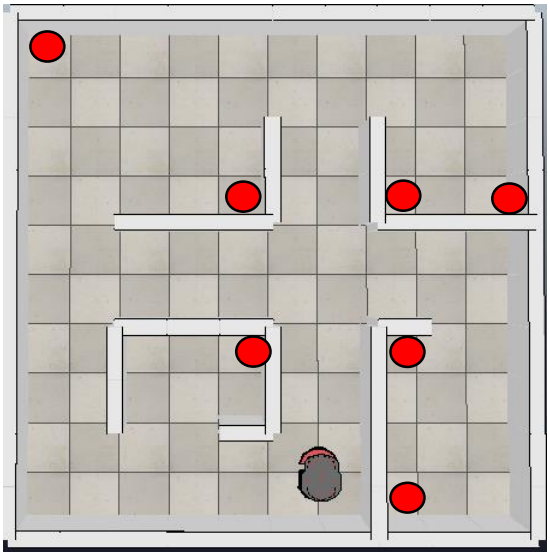
(Map 1)



(Map 2)



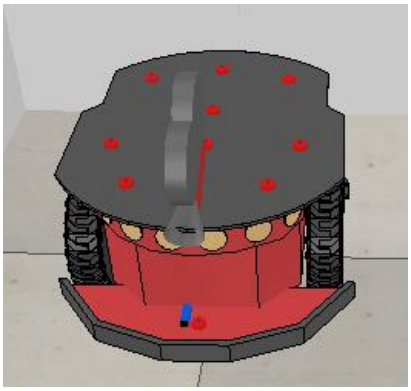
(Map 3)

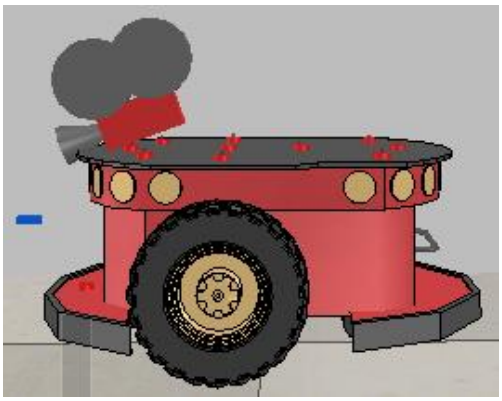


(Map 4)

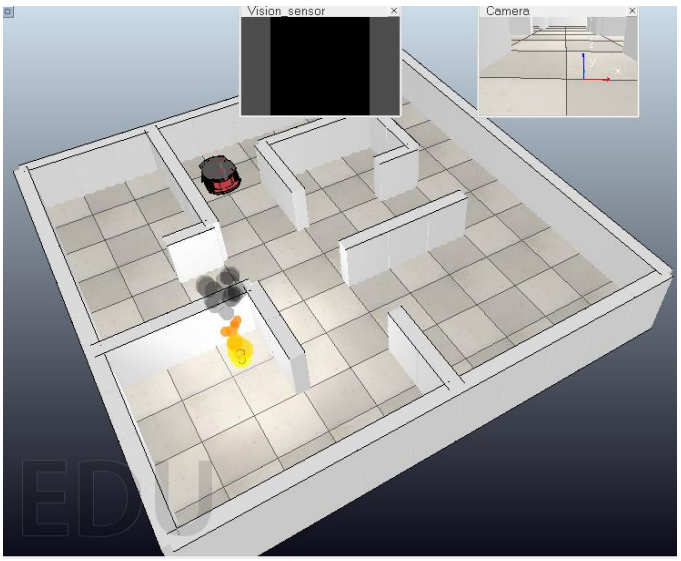
II. Soal

Seperti halnya soal UTS, diberikan suatu rumah yang mana terdapat api di dalamnya. Karena rumah ini tergolong rumah yang moderen dan canggih, maka konfigurasi ruangannya pun bisa berubah-ubah. Totalnya, ada empat kemungkinan konfigurasi. Keempat konfigurasi rumah tersebut dapat dilihat pada gambardi atas. Titik-titik berwarna merah adalah posisi-posisi yang mungkin terdapat api (Hanya ada satu api). Pada tugas kali ini, robot yang digunakan adalah robot pioneer, seperti yang ditampilkan gambar berikut.





Tugas robot pioneer adalah menjelajahi ruangan (maze), menemukan api dan mematikannya. Pada template soal, sudah disediakan map dan robot pioneer yang dilengkapi sensor ultrasonik, sensor kamera dan vision, serta denah ruangan, serta denah ruangan. Berikut tampilan salah satu map. Untuk mengubah dari satu map ke map lain dapat dilakukan dengan modifikasi lokasi (Wall 49,Wall50, Wall57,Wall58,Wall59).



Adapun tugas yang harus kalian selesaikan adalah sebagai berikut :

<u>Tugas A : Monte Carlo Localization (MCL):</u>

1. Tambahkan (pasang) aktuator berupa air jet atau sejenisnya pada robot untuk keperluan memamdamkan api saat robot sudah menemukan api. Berikut adalah gambar air jet

2. Rancang dan implementasikan algoritma untuk menemukan api untuk Salah satu Map. Setiap kelompok hanya mengerjakan satu map saja. Berikut adalah pembagian map per kelompok :

No	Kelompok	Map
1	A	Map 1
2	B	Map 2
3	C	Map 3
4	D	Map 4

   Implementasikan algoritma tersebut dalam kode python menggunakan teknik remote API seperti tutorial 1.
3. Implementasikan metode Standard MCL dan KLD-MCL selama robot bergerak mencari api. Silakan gunakan template yang disediakan dan pelajari tutorial 2 untuk mempermudah pengerjaan anda. Plot partikel setiap iterasi seperti pada soal praktikum MCL.
4. Eksperimen, bandingkan dan analisis ketepatan / akurasi metode Standard MCL dan KLD-MCL pada beberapa skenario percobaan, misalnya pada tabel berikut :

   Skenario 1 :

Iterasi	Error Standad MCL	Error KLD-MCL
1		
2		
3		
.....		

   Skenario 2 :

Iterasi	Error Standad MCL	Error KLD-MCL
1		
2		
3		
.....		

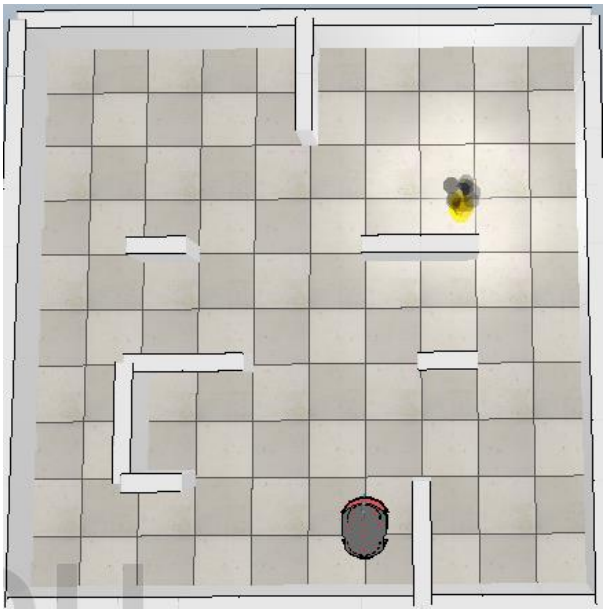
   Skenario 3, Skenario 4, dst

5. Komponen untuk MCL localization yang perlu diimplementasikan adalah sebagai berikut :
    a) **Robot simulation di VREP dan remote API python (direkomendasikan menggunakan python XY)**
    b) **Gridmap representation on python code (Silakan modelkan lapangan dengan model grid seperti tutorial MCL [ukuran grid bebas])**
    c) **Velocity Motion Model (motion model)**
    d) **Beam Model for range finders (perception model)**
    e) **Standard MCL**
    f) **KLD-sampling MCL**

6. Buatlah laporan dokumentasi pengerjaan tugas tersebut.

<u>Tugas B : Planning under Markov Decision Processes (MDPs) :</u>

1. Modelkan lapangan sebagai grid seperti halnya soal A
2. Modelkan perception model, motion model sperti halnya Soal A
3. Diberikan posisi awal robot dan posisi api sesuai pada Scene soal B. Implementasikan MDP agar robot menuju posisi api secepat mungkin.



4. Implementasikan hasil planning point 3 pada simulator V-Rep dengan remote API python seperti soal A.

III. Laporan dan Presentasi

1. Tulislah laporan untuk kedua soal di atas dalam format artikel ilmiah seperti pada template.
2. Buatlah presentasi dan video untuk menunjang demo

IV. Penilaian

Berikut ada;ah komponen penilaian tugas UAS kelompok :

- 15% Penyelesaian algoritma pencarian api
- 40% Implementasi Soal A : MCL
- 30% Implementasi Soal B : MDP
- 25% Laporan dan presentasi

Bonus (max 15%) :

- Implementasikan sistem ini menggunakan Platform Ubuntu dan ROS.

V. Pengumpulan

Kumpulkan pengerjaan tugas kelompok anda dengan format “UAS_kelompok[nomor].zip” yang berisi file program, scene v-rep, laporan/dokumentasi, dan presentasi. Keterlambatan pengumpulan akan mendapat pengurangan 5 poin setiap hari.

VI. Presentasi, Demo dan Tanya Jawab

Setiap kelompok akan mendapatkan waktu total untuk presentasi, demo, dan tanya jawab minimal 30 menit.

Robotika 2016/2017 **V.02-20170403**

Tugas Pengganti UTS

Dosen: Dr.Eng. Wisnu Jatmiko, M.Kom; M. Anwar Ma'sum, M.Kom

Asisten: Novian Habibie, Aitya Murda Nugraha

1.1 Pendahuluan

Tugas Pengganti UTS mata kuliah robotika mengangkat tema navigasi sederhana pada simulator Gazebo menggunakan Robot Operating System (ROS).

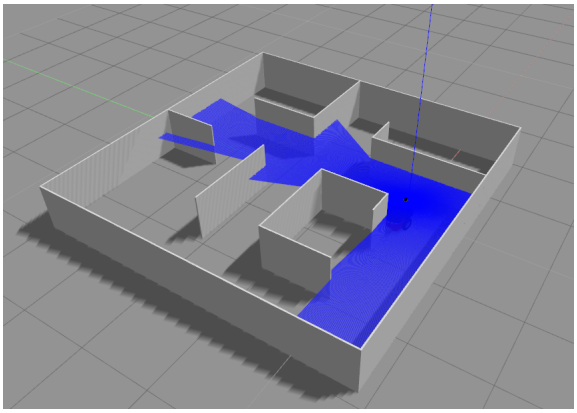


Figure 1.1: Gazebo dan ROS

1.1.1 Deskripsi Soal

Pada tugas kali ini, terdapat sebuah *world* yang di dalamnya terdapat sebuah robot Pioneer di dalam gedung dengan 4 ruang. Pada masing-masing ruang terdapat api yang harus dipadamkan oleh robot.

1.1.1.1 Robot

Robot yang digunakan untuk tugas kali ini adalah robot Pioneer P2DX. Pada robot terdapat banyak sensor yang dapat digunakan. Pada *template* yang diberikan robot dilengkapi dengan sensor *laser*. Pada ROS, nilai laser dapat diakses pada *topic* /base_scan dengan tipe data /sensor_msgs/LaserScan.

Pada robot yang digunakan, terdapat 2 buah *motor* sebagai *actuator* pada roda. *Motor* pada roda dapat dikendalikan dengan memberikan pesan pada *topic* /pioneer/cmd_vel.

1.1.1.2 Gedung

Simulasi gedung yang digunakan adalah peta yang serupa pada Trinity College Fire-Fighting Home Robot Contest (TCCFHRC) (`http://www.trincoll.edu/events/robot/about-us.html`, *rules* dapat diunduh di

`http://ecesd.engr.uconn.edu/ecesd1617/files/2015/09/TCFFHRC_Rules2016_099.pdf`) Pada gedung terdapat 4 buah ruangan yang di dalamnya terdapat titik api yang harus dapat dipadamkan oleh robot. Di antara ruang terdapat koridor yang dapat dilalui robot. Berikut ini adalah peta dari gedung.

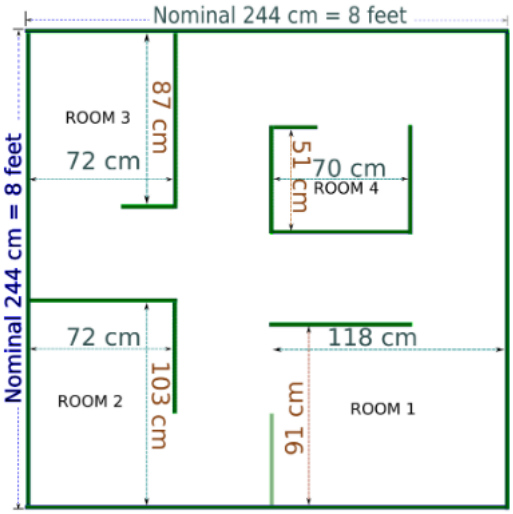


Figure 1.2: Peta dari gedung.

Terdapat 4 ruangan pada gedung yang pada masing-masing ruangan terdapat titik api. Masing-masing api dapat berada di titik manapun di dalam ruangan. Robot terlebih dulu harus mendeteksi letak api di dalam masing-masing ruang dan mendekati api, baru kemudian menyalakan kipas untuk memadamkan api.

1.2 Tugas

Misi besar dari tugas ini adalah melakukan navigasi menggunakan robot Pioneer-P2DX dan memadamkan api. Tugas dibagi menjadi dua bagian, dimana setiap bagian menggunakan *template* yang berbeda

1. Implementasi Algoritma *Wallfollowing* (Minggu Pertama) Implementasi navigasi sederhana pada robot dengan pendekatan *wall follwing* (bergerak meyusuri tembok) dengan menanfaatkan sensor jarak (*laser*) dan pergerakan robot secara linear dan angular. Template pertama hanya berisikan ruangan tanpa api dan garis pembatas pintu.

2. Navigasi Ruangan dan Deteksi Api (Minggu Kedua) Setelah algoritma *wallfollowing* berhasil diimplementasikan pada robot, implementasikan algoritma navigasi yang efisien sehingga dapat berpindah ke ruangan lain dalam waktu singkat dan dapat mendeteksi adanya api pada ruangan yang memiliki objek api. Template kedua dilengkapi dengan garis pembatas pintu dan objek api.

1.3 Penilaian

Beberapa aspek yang harus dipenuhi pada tugas ini antara lain:

1. Bagian 1
   (a) Robot dapat membaca jarak menggunakan sensor *laser*
   (b) Robot dapat bergerak menggunakan aktuator dengan acuan pembacaan sensor jarak
   (c) Robot dapat masuk ke ruangan 1, 2, dan 3
   (d) (BONUS) Mengimplementasikan algoritma PID (`https://en.wikipedia.org/wiki/PID_controller`) untuk *wallfollowing*
2. Bagian 2
   (a) Robot dapat mendeteksi garis pintu ruangan
   (b) Robot dapat mendeteksi api pada ruangan yang memiliki objek api
   (c) Robot dapat masuk ke ruangan 4
   (d) (BONUS) Robot dapat kembali ke posisi *start*
   (e) (BONUS) Robot dapat *start* dari dalam ruangan

1.3.1 [UPDATED]Konten Laporan

(a) Pendahuluan singkat
(b) Penjelasan algoritma dan metodologi yang digunakan pada robot
(c) Eksperimen, berupa
   i. Waktu tempuh robot ke tiap ruangan
   ii. Akurasi pendektesian api
   iii. Kestabilan *wallfollowing*
(d) Analisis hasil eksperimen
(e) Kesimpulan

1.4 Deadline dan Pengumpulan

Deadline dibagi menjadi dua gelombang:

1. Bagian 1 : **Jumat, 31 Maret 2017 (23:55 WIB)** Kode + laporan + *progress report* ke tim asisten pada tanggal 3 April 2017
2. Bagian 2 : **[UPDATED]Minggu, 9 April 2017 (23:55 WIB)** Kode + laporan + presentasi

[UPDATED]Demo tugas dilaksanakan pada hari Senin, 10 April 2017. Demo tugas dilaksanakan selama 30 menit per kelompok, terdiri atas presentasi singkat, demo program, dan tanya jawab. Slot demo tugas akan dibuka pada forum tugas pengganti UAS `https://scele.cs.ui.ac.id/mod/forum/view.php?id=8850`

Selamat Mengerjakan!

Robotika 2016/2017 **V.02-20170602**

Tugas Pengganti UAS

Dosen: Dr.Eng. Wisnu Jatmiko, M.Kom; M. Anwar Ma'sum, M.Kom

Asisten: Novian Habibie, Aditya Murda Nugraha

0.1 Pendahuluan

Tugas Pengganti UAS mata kuliah robotika mengangkat tema lokalisasi pada simulator Gazebo menggunakan Robot Operating System (ROS) dengan *environment* kebun apel.

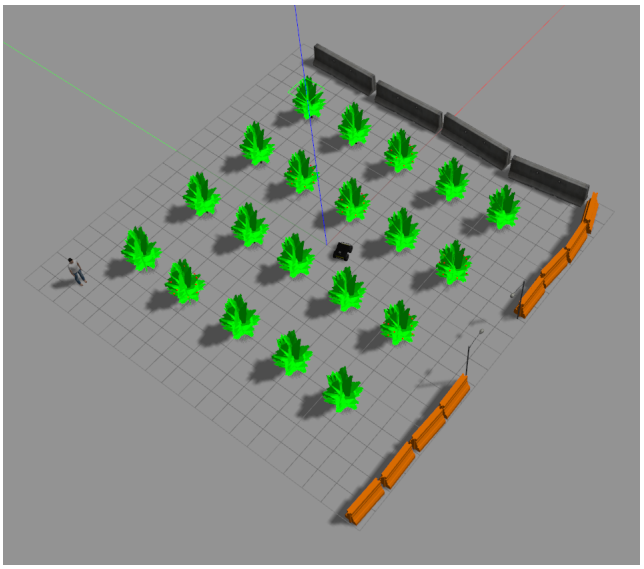


Figure 0.1: *Environment* kebun apel pada Gazebo

0.1.1 Deskripsi Soal

Untuk tugas pengganti UAS, *scene* yang digunakan adalah simulasi perkebunan apel. Pada kebun apel terdapat pohon apel, dinding pembatas kebun, dan beberpa objek tambahan seperti manusia dan lampu. Tujuan utama dari tugas ini adalah menjalankan Monte Carlo Localization (MCL) yang dapat memprediksi posisi robot pada peta kebun apel.

0.1.1.1 Robot

Robot yang digunakan untuk tugas kali ini adalah robot Husky buatan Clearpath Robotics. Husky adalah *unmanned ground vehicle* yang didesain untuk eksplorasi luar ruangan *outdoor*. Robot ini bergerak menggunakan 4 roda dengan prinsip gerak *differential drive*. Dalam melakukan navigasi, robot dapat dikustomisasi dengan memasangkan berbagai macam tipe sensor, salah satu sensor yang umum digunakan bersama robot

Husky adalah sensor laser 2D lIDAR LMS1xx buatan SICK. Detail mengenai robot Husky dapat dilihat di `http://wiki.ros.org/Robots/Husky` dan LMS1xx dapat dilihat di `http://wiki.ros.org/LMS1xx`.

0.1.1.2 Kebun Apel

Pada tugas pengganti UAS, *environment* yang digunakan adalah simulasi kebun apel pada simulator gazebo, terdiri atas model pohon apel, pembatas kebun, objek lampu dan manusia. Contoh visual dari kebun apel yang digunakan dapat dilihat pada gambar 1.1.

0.2 Tugas

Misi besar dari tugas ini adalah melakukan estimasi lokasi (*localization*) robot pada kebun apel. *Localization* dilakukan dengan mengimplementasikan *motion model* dan *sensor model* robot kemudian menggunakan keduanya pada algoritma Monte Carlo Localization menggunakan Particle Filter. *Localizaton* dilakukan dengan acuan peta kebun apel yang sudah disediakan.

1. Implementasi *motion model*

   Motion model yang digunakan adalah *velocity-based motion model*, dikarenakan untuk menggerakkan robot Husky melalui ROS masukan yang diterima adalah kecepatan angular dan linear robot. Luaran yang diharapkan dari *motion model* adalah estimasi *pose* robot (x, y, θ) terhadap pose awal setelah diberikan kecepatan linear (v_l) dan kecepatan angular (v_a).

2. Implementasi *sensor model*

   Robot Husky menggunakan sensor laser yang mendapatkan jarak terhadap objek di sekeliling robot untuk rentang pembacaan 270 derajat dan membaca sekitar 16000 titik. Untuk mempermudah implementasi *sensor model*, arah pembacaan yang digunakan mengacu pada 16-Wind Compass Rose (gambar 1.2), yaitu 16 arah *data sampling*. Dikarenakan rentang pembacaan laser yang hanya 270 derajat menyebabkan titik SSW, S, SSE tidak terbaca, maka jumlah minimal arah yang digunakan adalah 13 titik.

   Catatan : untuk 1 arah *sampling* dapat menggunakan beberapa titik yang berdekatan kemudian dirata-ratakan nilai pembacaanya.

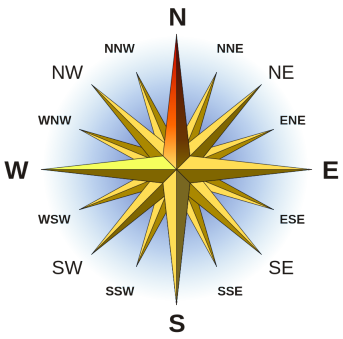


Figure 0.2: 16-Wind Compass Rose

3. Implementasi Particle Filter untuk Mote Carlo Localization

   Momte Carlo Localization dilakukan menggunakan kumpulan partikel yang disebar pada peta untuk kemudian dilakukan iterasi yang akan mengkonvergenkan partikel yang ada menuju satu titik estimasi lokasi. *Localization* yang digunakan menggunakan Particle Filter. Peta lingungan sudah disediakan dan tidak perlu dibuat lagi.

0.3 Penilaian

Beberapa aspek yang harus dipenuhi pada tugas ini antara lain:

1. Implementasi *motion model* (10%)
2. Implementasi *sensor model* (10%)
3. Implementasi Monte Carlo Localization (30%)
4. Eksperimen dan analisis (20%)
5. Laporan dan dokumentasi (10%)

0.4 Deadline dan Pengumpulan

Kamis, 8 Juni 2017, 09:00 WIB

Kumpulkan kode program (tanpa file robot/simulasi) + dokumen laporan + file presentasi.

Demo dilakukan di hari yang sama, slot demo menyusul.

Selamat Mengerjakan!

0.5 Langkah-Langkah Pengerjaan

0.5.1 Instalasi dan Uji Coba *Environment*

0.5.1.1 Instalasi Robot

Robot yang digunakan adalah robot Husky versi terbaru. Untuk menginstall, buat sebuah *catkin workspace* terlebih dahulu, lalu clone file-file Husky dari Github `https://github.com/husky/husky` pada folder *src*.

```
git clone https://github.com/husky/husky
```

Husky versi terbaru menggunakan LMS1xx versi terbaru, yang pada saat ini masih belum tersedia di ROS. Untuk itu, siapkan juga LMS1xx terbaru melalui *source*, dengan clone dari Github `https://github.com/clearpathrobotics/LMS1xx`.

```
git clone https://github.com/clearpathrobotics/LMS1xx
```

Setelah semua kelengapan tersedia, *compile* kode robot Husky dan LMS1xx pada *catkin workspace* anda dengan menjalankan perintah

```
catkin_make
```

Jika terdapat *error* ketika proses *compile*, kemungkinan ada beberapa *package* pada ROS anda yang rusak/tidak terinstall dengan baik. Untuk memperbaikinya, jalankan perintah berikut :

```
rosdep install --from-paths src --ignore-src --rosdistro kinetic -y -r
```

Jika proses kompilasi sudah berjalan dengan baik, uji coba robot mengikuti tutorial di `http://wiki.ros.org/husky_navigation/Tutorials`. Namun ingat, sebelum menjalankan program ROS apapun pada sebuah terminal baru, *source* folder *catkin workspace* tempat robot Husky dan sensor LMS1xx berada dengan menjalankan perintah :

```
source ./devel/setup.bash
```

0.5.1.2 Instalasi *World* Kebun Apel

Unduh *file* world yang telah disediakan di Scele, lalu simpan pada direktori yang sama dengan Husky *huskyhusky_gazeboworlds*.

Untuk memanggil file *world*, dibutuhkan suatu *descriptor* untuk *world* tersebut. Unduh file URDF yang disediakan, dan tempatkan di direktori yang sama dengan file .world.

Setelah file *world* dan URDF dibuat, langkah selanjutnya adalah menguji coba pemanggilan kebun apel, tanpa robot terlebih dulu. Jalankan Gazebo dengan perintah

```
rosrun gazebo_ros world/appleworld.world
```

dan perhatikan apakah *world* kebun apel dapat muncul. Jika berjalan dengan benar maka kebun apel akan muncul. Jika kebun apel tidak muncul maka ada kesalahan. Periksa kembali konfigurasi dan lokasi file dan folder.

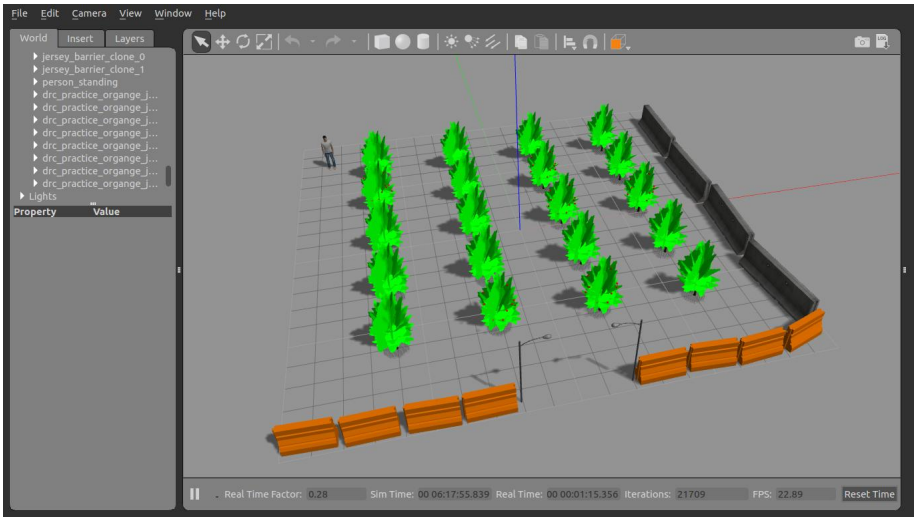


Figure 0.3: Contoh kebun apel

0.5.2 Integrasi Husky dengan Kebun Apel

Setelah file *world* dan URDF ditempatkan, langkah selanjutnya adalah menambahkan file *launcher* pada folder husky_gazebo/launch. Buat sebuah file baru, misalkan diberi nama applepen.launch, dan isikan kode berikut ini.

```
<launch>

  <arg name="world_name" default="worlds/empty.world"/>

  <arg name="laser_enabled" default="true"/>
  <arg name="kinect_enabled" default="false"/>

  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="$(find husky_gazebo)/worlds/applefarm2b.world"/>
    <arg name="paused" value="false"/>
    <arg name="use_sim_time" value="true"/>
    <arg name="gui" value="true"/>
    <arg name="headless" value="false"/>
    <arg name="debug" value="false"/>
  </include>

  <include file="$(find husky_gazebo)/launch/spawn_husky.launch">
```

```
    <arg name="laser_enabled" value="$(arg laser_enabled)"/>
    <arg name="kinect_enabled" value="$(arg kinect_enabled)"/>
    <arg name="x" value="8"/>
    <arg name="y" value="6"/>
    <arg name="yaw" value="-1.55"/>
  </include>

</launch>
```

Sesuaikan lokasi-lokasi file dan direktori dengan pengaturan yang Anda lakukan. Setelah *load* kebun berhasil, langkah selanjutnya adalah menguji coba pemanggilan Husky dan *world* baru. Pada *launcher* di atas, kita sudah memasukkan definisi-definisi dan file-file yang diperlukan. Jalankan

```
roslaunch gazebo_world applepen.launch
```

untuk memanggil Gazebo dengan *world* dan robot yang baru. Jika berjalan dengan benar, robot dan kebun apel akan tampil.

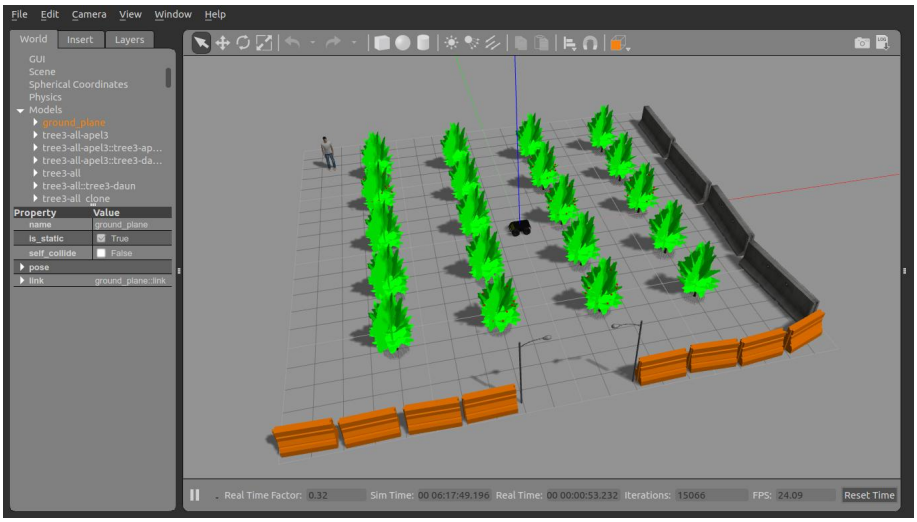


Figure 0.4: Contoh kebun apel dan robot

0.5.3 Menambahkan *Map* yang Sudah Dibangkitkan Sebelumnya

Pada tugas kali ini terdapat *map* yang sudah dibangkitkan sebelumnya untuk keperluan navigasi dan lokalisasi. *Map* disimpan dengan format PGM dan dapat diunduh di SCeLE. Tempatkan file PGM dan YAML dalam 1 folder. Untuk menggunakan *map* tersebut, pada file *launcher* tambahkan baris berikut.

```
<!-- <node pkg="map_server" type="map_server" args="\$(find
<<PACKAGE_PETA>>)/map.yaml" respawn="true" name="map1"/> -->
```

Ganti ¡¡PACKAGE_PETA¿¿map.yaml dengan lokasi file YAML yang diunduh. *Map* tersebut akan di-*broadcast* pada topik /map.

0.5.4 Implementasi Monte Carlo Localization

Buat sebuah *catkin workspace* yang baru, lalu buat package baru (`http://wiki.ros.org/ROS/Tutorials/catkin/CreatingPackage`) dengan dependensi *std-msgs, rocpp/rospy.* Buat kode program untuk *motion model, sensor model,* dan *particle filter* dalam dalam file terpisah yang akan berperan sebagai *node* untuk masing-masing algoritma (lihat cara membuat *node* pada tugas pengganti UTS).

Setiap *node* memerlukan *subscribe* ke *topic* tertentu sesuai kebutuhan. Untuk mengecek seluruh *topic* yang sedang tersedia, gunakan perintah:

```
rostopic list
```

Tutorial lengkap mengenai *rostopic* dapat dilihat di `http://wiki.ros.org/rostopic` .

Secara umum, alur implementasi dan komunikasi antar modul pada ROS untuk Monte Carlo Localization dapat dilihat pada gambar 0.5 :

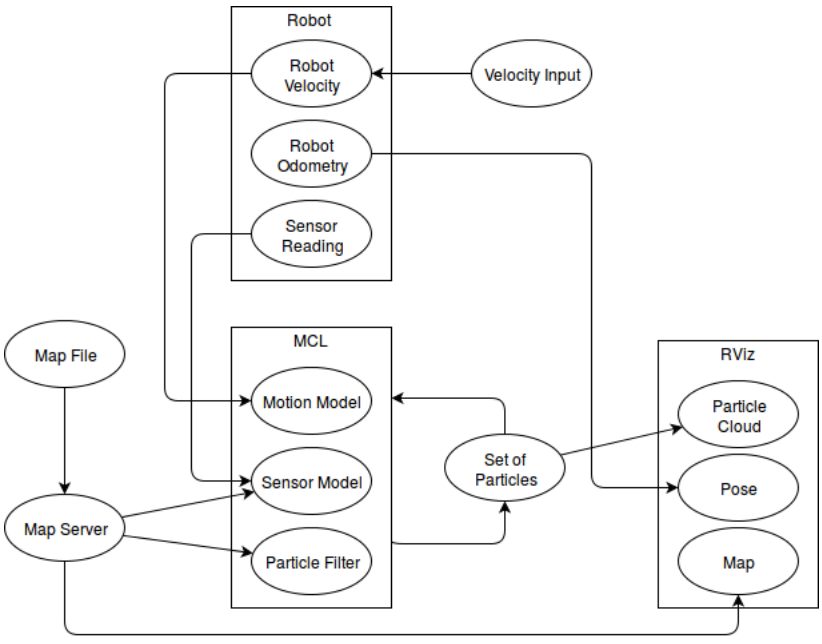


Figure 0.5: Alur MCL secara general

0.5.4.1 Motion Model

Buat sebuah fungsi Motion Model yang dapat menerima masukan dalam format kecepatan angular dan linear, dan memberikan luaran dalam bentuk *pose* robot dalam koordinat x, y dan sudut robot θ.

Untuk memberikan kecepatan linear dan angular pada robot Husky, anda dapat melakukan *publish* kepada *topic* **/husky_velocity_controller/cmd_vel** dengan cara yang sama seperti pada tugas UTS. Untuk mendapatkan *pose* robot, anda dapat melakukan *subscribe* ke *topic* **/husky_velocity_controller/odom**.

0.5.4.2 Sensor Model

Untuk mengakses data pembacaan sensor LMS1xx, lakukan *subscribe* ke *topic* **scan**. Lakukan pengambilan data dengan cara yang sama dengan tugas UTS.

Seperti yang sudah dijelaskan di kelas, sensor model adalah *set of probability* dari pembacaan sensor robot di tiap titik peta. Untuk membuat sensor model (khususnya untuk p_hit), lakukan sampling terhadap semua titik pada peta dengan menggunakan jumlah titik sampling minimal 13 arah mata angin (sudah dijelaskan di deskripsi soal). Lakukan sampling dengan memindahkan robot ke semua titik secara manual (atau secara otomatis menggunakan *script* yang anda buat) lalu disimpan dalam bentuk matriks 3 dimensi - x dan y untuk tiap koordinat yang disampling, z untuk data tiap arah mata angin.

Catatan :

1. Untuk membuat sensor model, anda tidak harus melakukan sampling pada tiap piksel/koordinat peta. Anda bisa mengasumsikan beberapa titik sebagai satu *grid* yang sama. Ukuran *grid* dapat anda tentukan sendiri. Perhatikan pula konsekuensi ukuran *grid*, semakin besar ukuran *grid* yang digunakan, semakin rendah akurasi *localization* yang anda buat.

2. Dalam proses sampling, asumsikan robot hanya menghadap ke satu arah saja. Untuk implementasi di Particle Filter nantinya, tentukan transformasi arah secara diskret (dengan mengacu kepada jumlah arah sampling dan perubahan derajat yang anda gunakan).

0.5.4.3 Representasi Partikel

Mari lihat terlebih dahulu contoh representasi *localization* yang telah disediakan ROS pada Husky. Jalankan tutorial `http://wiki.ros.org/husky_navigation/Tutorials/HuskyAMCLDemo`, lalu buka RViz yang telah berjalan. Pada RViz, buat sebuah visualisasi baru untuk partikel dengan klik **add** pada pojok kiri bawah. Pada *tab* **By display type**, pilih PoseArray, lalu klik OK. Setelah itu, jalankan robot dengan perintah **2D Nav Goal** yang ada di Menu Bar.

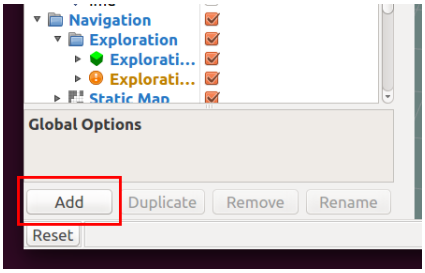


Figure 0.6: Klik Add untuk menambahkan visualisasi

Partikel pda RViz berbentuk kumpulan panah berwarna merah. Dikarenakan ukurannya kecil, saat konvergen partikel tertutupi oleh objek robot. Untuk melihat partikel, hilangkan sementara robot dari visualisasi dengan *deselect* RobotModel pada menu Displays.

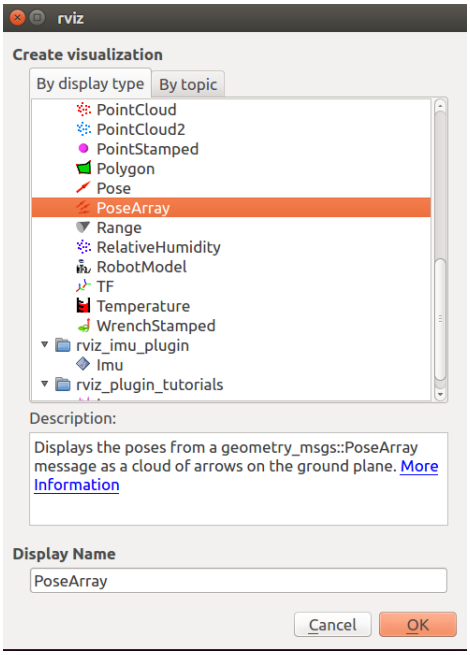


Figure 0.7: Pilih ParticleCloud

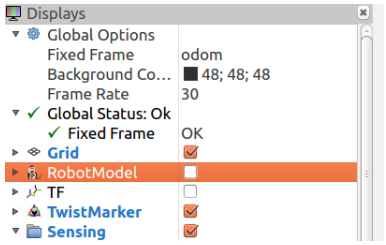


Figure 0.8: Hilangkan sementara objek robot

Setelah partikel terlihat, coba gerakkan robot menggunakan 2D Nav Goal mengelilingi peta, dan lihat bagaimana partikel menyebar dan mengumpul di lokasi-lokasi tertentu.

Representasi partikel pada RViz adalah visualisasi dari *topic* /particlelcoud, dimana *topic* tersebut adalah *array of pose* dari robot dengan tipe data geometry_msgs/PoseArray. Anda bisa melihat isi dari /particlecloud dengan menjalankan perintah *rostopic echo /particlecloud*).

0.5.4.4 Particle Filter

Pada implementasinya, Particle Filter akan mengolah setiap partikel menggunakan Motion Model dan Sensor Model, dan di tiap iterasinya akan menghasilkan partikel baru. Untuk membuat partikel baru, lakukan *publish* ke *topic* /particlecloud.

Catatan:

1. Untuk proses mengelilingi kebun, anda tidak perlu membuat navigasi anda sendiri. Silahkan gunakan navigasi yang sudah disediakan di Scele yang akan menggerakkan robot keliling kebun menggunakan

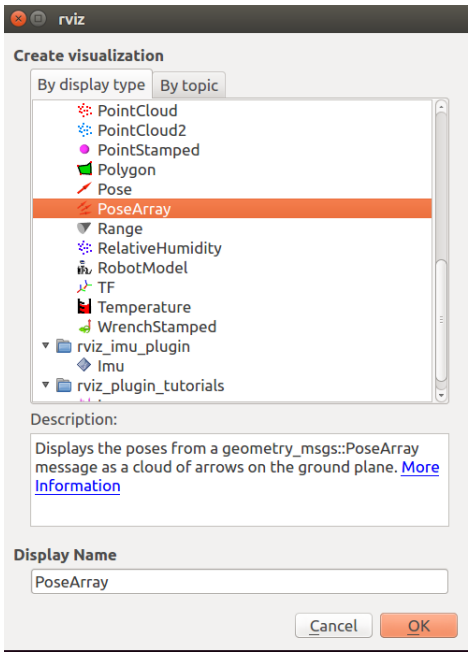


Figure 0.9: Gerakkan robot menggunakan 2D Nav Goal

wayopint. Baca `https://wiki.nps.edu/display/RC/Husky+Control+in+Gazebo` untuk cara penggunaan lebih lanjut.

2. Simpan partikel yang dibuat pada tiap iterasi dalam sebuah *array* untuk nantinya digunakan dalam pengukuran akurasi.

3. Walaupun MCL berjalan secara *real-time*, pada implementasinya δt yang digunakan tidaklah nol. Anda dapat mengatur δt sesuai kebutuhan anda, dengan konsekuensi penurunan akurasi saat δt bernilai besar.