

**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN LOKASI DAN
PEMETAAN SECARA SIMULTAN DENGAN OLAP**

Tesis

**Untuk memenuhi sebagian persyaratan
Mencapai derajat Sarjana S-2
Program Studi Magister Sistem Informasi**



**Disusun oleh :
Harindra Wisnu Pradhana
J4F009021**

**PROGRAM PASCA SARJANA
UNIVERSITAS DIPONEGORO
SEMARANG**

2013

SISTEM PENDUKUNG KEPUTUSAN PENENTUAN LOKASI DAN PEMETAAN SECARA SIMULTAN DENGAN OLAP

ABSTRAK

Penelitian ini mengulas *simultaneous localization and mapping* (SLAM) terpisah dari sistem robotik dengan memanfaatkan *data warehouse*. Pengolahan terpisah memungkinkan pengolahan kompleks dilakukan tanpa mengganggu kinerja sistem robotik. Sistem ini membantu menghasilkan dukungan keputusan dengan menganalisa informasi pada *data warehouse*. Informasi dukungan keputusan berupa visualisasi dalam grafik vektor untuk membantu interpretasi informasi robotik.

Sistem ini menggunakan *online analytical processing* (OLAP) pada proses pengolahan dan analisa informasi pada *data warehouse*. Proses visualisasi menggunakan *scalable vector graphics* (SVG) untuk menampilkan peta hasil pengolahan data robotik. *Web server* mengatur seluruh proses pengolahan dan analisa informasi OLAP dan menampilkannya melalui peramban pada pengguna.

Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang interaktif dan informatif. *Web server* dengan cepatnya mengolah informasi dan menampilkannya dalam browser. Pengolahan dan analisa dengan sistem ini menghasilkan informasi yang relevan dengan galat relatif kecil. Penelitian lebih lanjut dapat dilakukan dengan menambahkan metode-metode SLAM konvensional dalam mengolah informasi pada *data warehouse*.

Kata Kunci : Sistem Pendukung Keputusan, SLAM, *Data Warehouse*, OLAP, SVG.

DECISION SUPPORT SYSTEM OF SIMULTANEOUS LOCALIZATION AND MAPING USING OLAP

ABSTRACT

This research propose simultaneous localization and mapping (SLAM) system processed separately from the robot using dataware house. Information in the data warehouse processed separately to allow complex computation held without lowering robotic system performance. The system helps making decision by analyzing information in the data warehouse. Decision support information visualized in vector graphic to help robotic information interpretation.

This system use online analytical process (OLAP) on processing and analyzing information in the data warehouse. Visualization process use scalable vector graphics (SVG) to display map from robotic data processing. Web server manage all of the processing and analysis of OLAP information and displays it through browser to the user.

This research resulting interactive and informative decision support system. The web server quickly process information and displays it to the browser. System processing and analysis resulting relevant information with small error. For further research, conventional SLAM methods can be added to the server to operate the information in the data warehouse.

Keywords : Decision Support System, SLAM, Data Warehouse, OLAP, SVG.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Permasalahan SLAM (Simultaneous Localization and Mapping) dikemukakan pertama kali dalam IEEE *Robotics and Automation Conference* pada 1986 di San Francisco dimana pada masa itu metode-metode probabilitas statistik mulai dikenalkan pada robotika. Permasalahan SLAM yang diangkat adalah mungkin tidaknya sebuah robot bergerak secara otomatis dan secara bertahap menyusun sebuah peta sekaligus secara konsisten mendeteksi posisinya pada peta tersebut (Durrant-Whyte dan Bailey, 2006).

Dua permasalahan utama pada SLAM adalah lokalisasi robot secara relatif terhadap peta dan pemetaan berkesinambungan yang secara simultan melakukan pengkinian peta yang ada. Berbagai algoritma probabilitas terkait SLAM telah dikembangkan untuk menyelesaikan dua permasalahan tersebut dan secara teoritis permasalahan tersebut telah terselesaikan (Durrant-Whyte dan Bailey, 2006). Bagaimanapun juga, masih banyak peluang pengembangan system tersebut dengan pengayaan peta sebagai bagian dari algoritma SLAM.

Informasi pergerakan robotik dapat direpresentasikan dalam bentuk vektor untuk memudahkan proses penyimpanan dalam basis data (Pradhana dkk, 2013). Vektor dapat dengan mudah dioperasikan dan direkapitulasi untuk menentukan resultan dari beberapa vektor maupun selisih dari dua vektor. Konsep resultan vektor dapat digunakan pula untuk rekapitulasi posisi obyek secara relatif terhadap robot maupun terhadap titik acuan tertentu.

Sebuah sistem SLAM terintegrasi menghasilkan data gerakan dan deteksi berskala besar. Data deteksi dan gerakan SLAM yang sangat besar memerlukan sumber daya pengolahan yang besar pula. Pemecahan peta observasi berskala besar menjadi beberapa peta kecil membantu menyederhanakan pengolahan data dan mengurangi kerumitan sistem (Chong dan Kleeman, 1999). Proses

pengolahan data SLAM disimpan dalam beberapa peta lokal untuk kemudian dikonsolidasikan menjadi satu peta global.

Algoritma-algoritma SLAM pada umumnya ditanamkan pada *microcontroller* berukuran kecil yang berfungsi sebagai otak dari sebuah robot. Komputasi yang lebih rumit dapat dilakukan menggunakan sebuah komputer pada sistem terpisah (Pratama, 2013). Perkembangan teknologi komunikasi yang semakin pesat mendukung berbagai proses dilakukan dari jarak jauh. Sebuah informasi di satu tempat memungkinkan untuk diproses di tempat lain seketika data tersebut diterima. Hal ini sangat membantu pengembangan SLAM dengan menghilangkan keterbatasan fisik sebuah robot sehingga komputasi yang lebih rumit dapat dilakukan di luar fisik robot tersebut.

Data warehouse merupakan salah satu sistem yang mampu mengolah sekumpulan besar data dengan efektif dan efisien untuk menghasilkan informasi yang bermanfaat (Hammergren dan Simon, 2009). Informasi-informasi hasil dari pengolahan data *warehouse* dapat dianalisa untuk kemudian dijadikan dasar penentuan keputusan. Untuk menghasilkan informasi-informasi yang diperlukan, sebuah data *warehouse* perlu dilengkapi dengan *Online Analytical Processing* (OLAP) yang berfungsi menyusun data ke dalam matriks berdimensi tertentu yang kemudian dapat dimanfaatkan dalam menyelesaikan permasalahan SLAM. Dengan dukungan teknologi komunikasi, sebuah robot sederhana dapat menggunakan sebuah *data warehouse* dengan OLAP di tempat terpisah untuk menentukan dimana posisi saat ini secara relatif terhadap peta serta melakukan pengkinian peta yang ada.

Tesis ini meneliti pemanfaatan OLAP dalam pengelolaan data deteksi lingkungan dari sebuah robot sederhana. Data hasil deteksi tersebut dikemas dalam beberapa dimensi untuk dianalisa. Hasil analisa data tersebut menjadi informasi kemana robot bergerak dan posisi obyek-boyek yang ditemui kemudian serta melakukan pengkinian terhadap peta yang ada.

1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan sistem OLAP untuk mengolah data SLAM yang mampu menampilkan peta dengan pengkinian otomatis, mengetahui posisi terkini dari agen secara relatif terhadap peta, serta menginformasikan area-area potensial yang belum dipetakan untuk membantu agen menentukan keputusan kemana langkah eksplorasi selanjutnya.

1.3 Manfaat Penelitian

Manfaat dari penelitian ini antara lain :

1. Menunjang sistem kerja elektromekanik.
2. Membantu observasi obyek di sebuah area tanpa merusak kondisi yang ada (Non Destruktif).
3. Membantu menginterpretasikan visi robotik.
4. Membantu visualisasi kondisi dan posisi robotik.

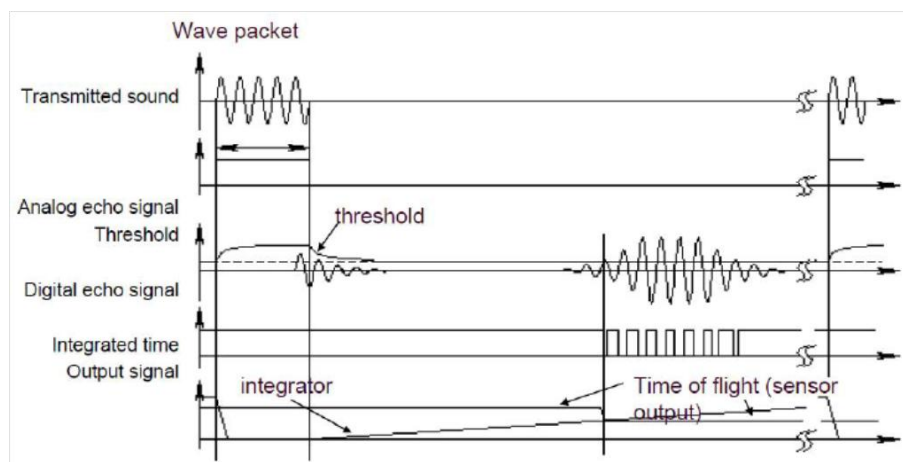
BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Meninjau berbagai penelitian seputar SLAM sebelumnya, telah dihasilkan beragam algoritma yang berhasil menyelesaikan berbagai permasalahan SLAM dengan karakteristik yang beragam. Setiap solusi yang ditawarkan suatu algoritma memiliki kelebihan dan kekurangan masing-masing. Penelitian-penelitian tersebut juga menyimpan dokumentasi berbagai perangkat yang digunakan pada robot baik sistem sensorik dan motorik.

2.1.1 Sensor Robotik

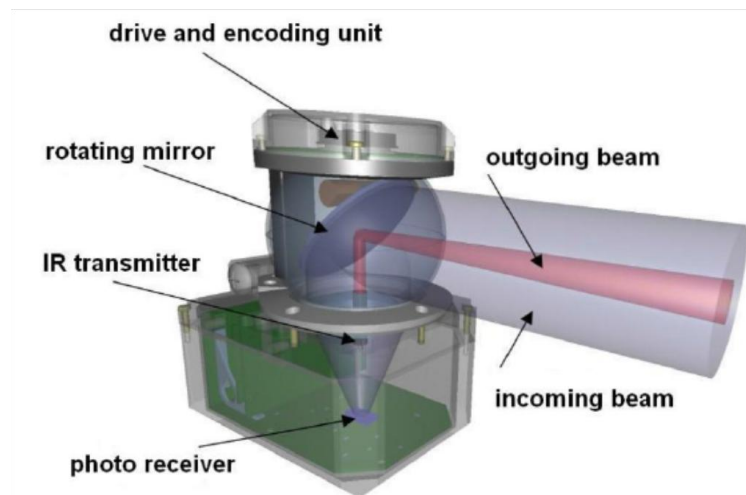
Sebuah sistem SLAM telah menggunakan sensor sonar (*Sound Navigation and Ranging*) dengan karakteristik relatif murah dan memiliki tingkat keakuratan deteksi antara 15cm hingga 5m (Ruckert, 2009). Prinsip kerja dari sonar adalah dengan memancarkan paket gelombang dan menunggu pantulannya, durasi antara gelombang dikirim dan diterima dapat digunakan untuk menghitung jarak obyek dari sensor. Prinsip kerja sonar ditampilkan pada gambar 2.1.



Gambar 2.1 Prinsip kerja sonar (Ruckert, 2009)

Laser Range Finder yang memiliki karakteristik paling mahal dengan

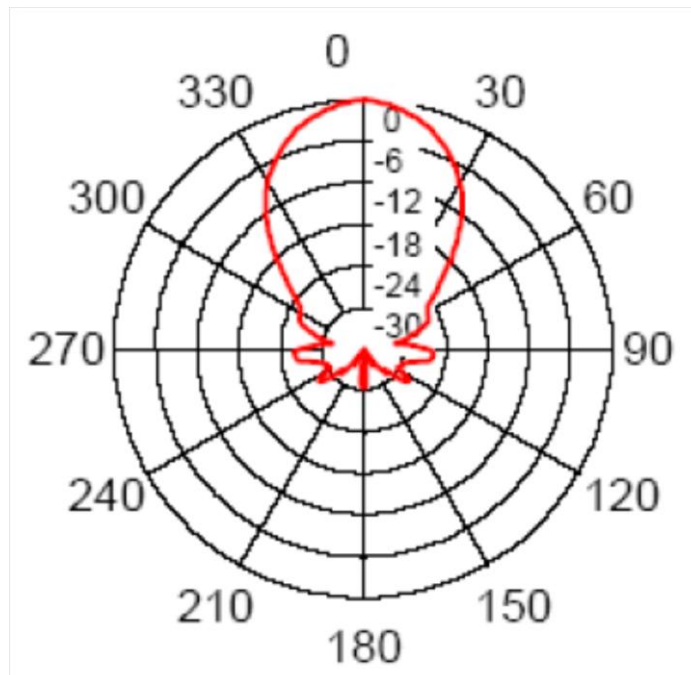
tingkat keakuratan paling tinggi dan dapat digunakan untuk deteksi hingga 200m (Ruckert, 2009). Prinsip kerja *laser range finder* relatif sama dengan sonar, hanya saja media yang digunakan adalah gelombang cahaya infra merah. Prinsip kerja sensor laser ditampilkan pada gambar 2.2.



Gambar 2.2 Prinsip sensor laser (Ruckert, 2009)

Sensor kamera memiliki berbagai fasilitas tambahan seperti sensor 3 dimensi yang dapat mendeteksi obyek dengan lebih presisi. Namun dalam implementasinya dengan SLAM, deteksi kamera memerlukan kalkulasi yang rumit untuk mengolah data citra ke jarak obyek dari robot sehingga sering kali hanya dipergunakan untuk sensor tambahan yang membedakan obyek satu dengan yang lain (Ruckert, 2009).

Sonar memiliki efektifitas tinggi untuk deteksi obyek dikarenakan memiliki desain yang kecil dan ringan sehingga mudah dipasang pada robot dan tidak memerlukan sumber tegangan khusus. Disamping itu sonar memiliki harga yang relatif murah dan mudah didapatkan. Berbeda dengan laser, sensor dengan sonar memiliki karakter yang unik karena bentuk fisik gelombang yang menyebar merata serta deteksi yang juga dari berbagai arah, sonar sangat mungkin menerima *noise* dari sekitarnya (Saleem, 2013). Visualisasi sensitifitas sonar ditampilkan pada gambar 2.3.



Gambar 2.3 Sensitifitas SONAR pada berbagai sudut deteksi (Saleem, 2013)

2.1.2 Motor Robotik

Robot yang diteliti memerlukan aktuator yang akurat dan presisi yaitu motor stepper. Motor stepper memiliki keakuratan tinggi dan respon yang sangat cepat sehingga efektif dipergunakan dalam berbagai sistem seperti periferil komputer, kendali proses, mesin-mesin industri dan robotik (Pal dan Tripathy, 2011).

Prinsip kerja motor stepper adalah mengatur beberapa pin input menjadi kutub-kutub magnet di dalam motor. Bila salah satu pin memiliki tegangan lebih tinggi, maka pin tersebut mengaktifkan kutub utara di dalam motor dan kutub-kutub yang lain menjadi selatan. Posisi kutub yang ada mengakibatkan magnet permanen pada rotor mengarah ke kutub yang berlawanan. Perubahan kutub-kutub berdasarkan tegangan pin akan memutar rotor pada motor stepper. Pada motor stepper, dalam mengendalikan rotor bisa dilakukan langkah yang lebih presisi dengan kombinasi satu kutub dan dua kutub bergantian (Singh dkk, 2010). Siklus

langkah-langkah tegangan input pin pada motor *stepper* dijelaskan pada tabel 2.1.

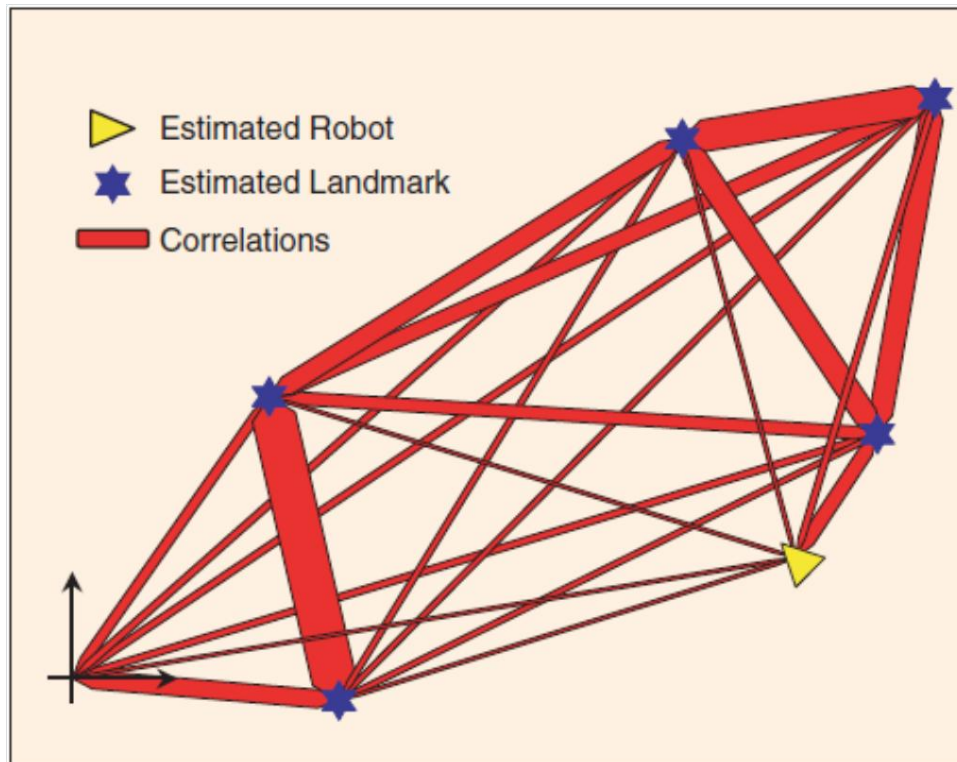
Tabel 2.1 Siklus pin motor stepper (Singh dkk, 2010)

Step	Pin 1	Pin 2	Pin 3	Pin 4
1	High	Low	Low	Low
2	High	High	Low	Low
3	Low	High	Low	Low
4	Low	High	High	Low
5	Low	Low	High	Low
6	Low	Low	High	High
7	Low	Low	Low	High
8	High	Low	Low	High

2.1.3 SLAM

Salah satu hal yang mendasar pada SLAM adalah dengan beredarnya robot pada area yang dieksplorasi, ditemui besaran korelasi antara obyek yang satu dengan yang lain bergeser sedikit demi sedikit ke sebuah nilai yang semakin akurat. Korelasi antar obyek yang nilainya berkembang seiring pergerakan robot ini dapat di analogikan sebagai pegas dengan nilai dinamis yang dengan kelenturan tertentu (Durrant-Whyte dan Bailey 2006). Untuk merepresentasikan kondisi ini dapat dibuat peta pegas yang menggambarkan posisi tiap-tiap obyek pada peta serta korelasi antar obyek yang satu dengan yang lain termasuk korelasi antara obyek dengan robot. Bentuk representasi relasi antar obyek pada peta pegas ditampilkan pada gambar 2.4.

Dalam proses deteksi lingkungan, jumlah obyek di sekitar robot terus berkembang dengan adanya deteksi obyek-obyek baru, metode submap membantu memberi ruang untuk menggambarkan kondisi area yang dipetakan (Bailey dan Durrant-Whyte, 2006).

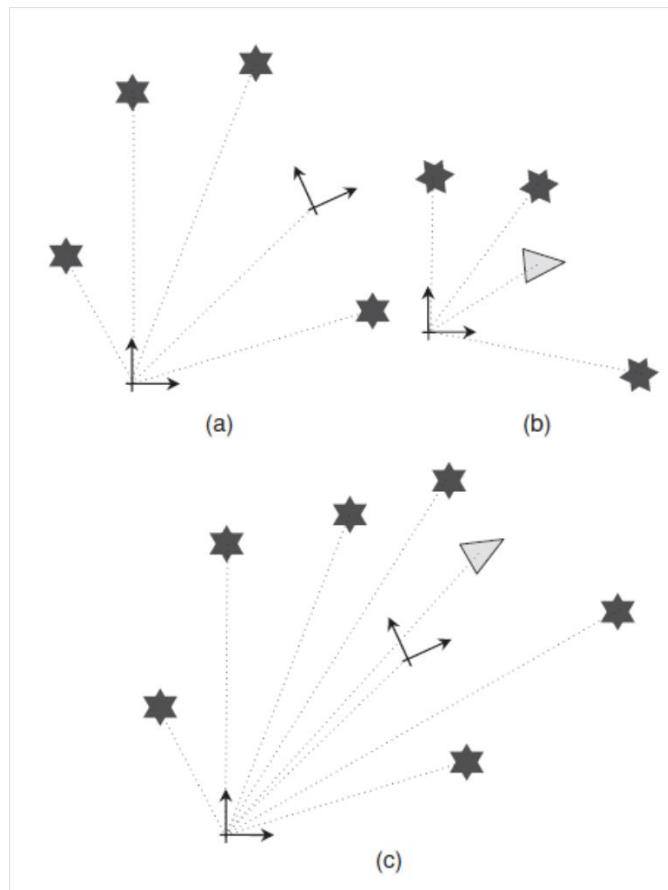


Gambar 2.4 Analogi SLAM dengan peta pegas (Durrant-Whyte dan bailey, 2006)

Untuk efisiensi proses sebuah peta dapat diuraikan menjadi beberapa peta yang lebih kecil (Chong dan Kleeman, 1999), pembagian ini mengelompokkan peta menjadi dua diantaranya adalah :

- Peta Global, adalah peta secara umum yang telah terdeteksi atau diketahui sebelumnya. Berisi seluruh obyek yang telah diketahui posisinya.
- Peta Lokal, adalah peta yang disimpan oleh robot ketika melakukan deteksi, bisa berisi obyek-obyek baru maupun lama yang sudah ada di peta global. Obyek lama ini kemudian dibandingkan dengan peta global sehingga dapat mengkonsolidasikan peta lokal untuk memperkaya peta global.

Bentuk tampilan peta lokal dan peta global serta hasil konsolidasinya ditampilkan pada gambar 2.5.



Gambar 2.5 Contoh registrasi peta lokal (b) ke dalam peta global (a) sehingga menghasilkan peta global terkini (c) (Bailey dan Durrant-Whyte, 2006)

2.1.4 Data Warehouse dan OLAP

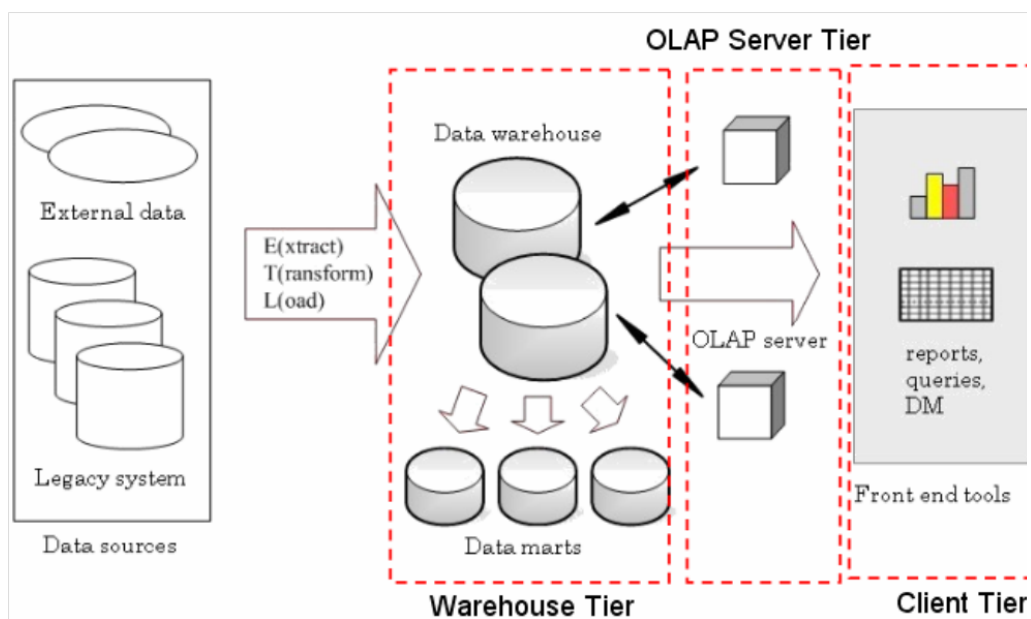
OLAP dalam fungsinya menganalisa *data warehouse* memiliki kemampuan mengeksplorasi data multi dimensi dan menampilkannya secara lengkap dan terstruktur (Bimonte dan Tchounikine, 2007). Data disusun kedalam kubus berdimensi n yang merepresentasikan komponen-komponen informasi yang hendak dianalisa. Informasi pada OLAP disusun dari data-data yang ada pada *data warehouse* dengan berbagai metode *query*. Secara umum *data warehouse* disusun dalam tiga lapis diantaranya adalah sebagai berikut :

- *Warehouse tier*, pada lapisan ini seluruh data dikumpulkan dan di

proses dengan berbagai operasi untuk menghasilkan informasi yang seragam.

- *OLAP tier*, pada lapisan ini komponen-komponen informasi disiapkan untuk menyusun kubus informasi yang siap dianalisa.
- *Client tier*, lapisan ketiga merupakan sisi dari sistem yang berhubungan langsung dengan pengguna. Pada lapisan ini informasi divisualisasikan agar mudah dibaca dan dipahami pengguna.

Ilustrasi arsitektur *data warehouse* dan OLAP ditampilkan pada gambar 2.6 berikut.



Gambar 2.6 Arsitektur data warehouse dengan OLAP (Bimonte dan Tchounikine, 2007)

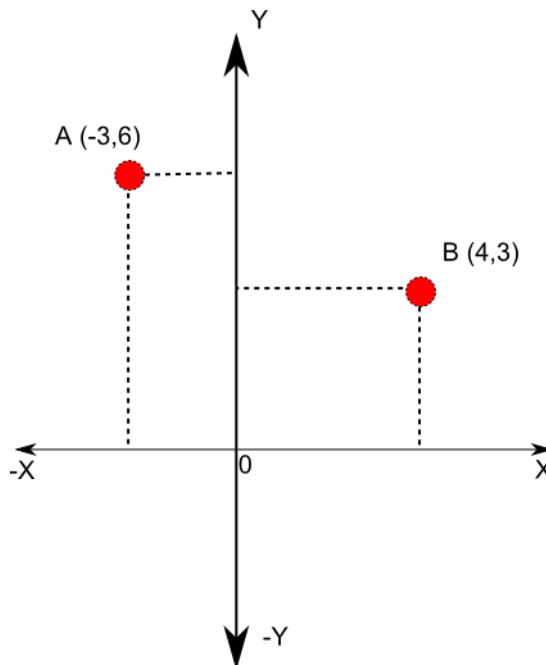
2.2 Dasar Teori

Pada bagian ini diulas beberapa teori-teori dasar yang mendukung proses penelitian. Teori-teori dasar tersebut meliputi algoritma-algoritma yang digunakan dalam mengoperasikan informasi SLAM hingga metode-metode representasi informasi tersebut dalam *data warehouse* dan OLAP. Penjabaran lebih lanjut disampaikan pada uraian berikut.

2.2.1 Sistem Koordinat

Penelitian ini merepresentasikan posisi-posisi obyek pada bidang 2 dimensi. Untuk menyimpan informasi posisi obyek pada suatu bidang 2 dimensi ada berbagai cara, salah satu cara yang umum digunakan adalah sistem koordinat kartesian. Sistem koordinat kartesian menyimpan informasi jarak horizontal dan vertikal suatu obyek dengan titik acuan tertentu (Strang, 1991).

Notasi dari sistem koordinat kartesian adalah jarak horisontal dan jarak vertikal dengan pemisah tanda koma dikemas dalam kurung sebagai contoh (x,y) (Strang, 1991). Jarak horisontal ke kiri dinotasikan sebagai bilangan negatif sedangkan jarak horisontal ke kanan dinotasikan sebagai bilangan positif. Jarak vertikal ke atas dinotasikan sebagai bilangan positif sedangkan jarak vertikal ke bawah dinotasikan sebagai bilangan negatif. Contoh representasi posisi pada koordinat kartesian ditampilkan pada gambar 2.7.

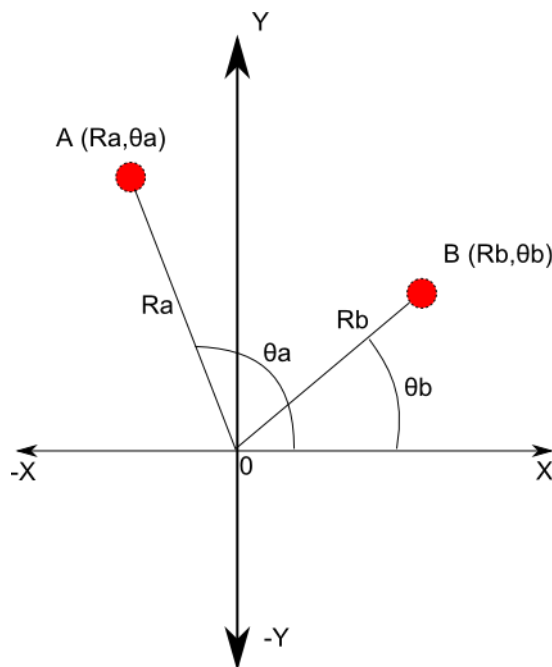


Gambar 2.7 Sistem koordinat kartesian

Gambar 2.7 menampilkan contoh representasi dua buah obyek pada bidang 2 dimensi. Obyek A pada gambar memiliki jarak horisontal sebesar 3 satuan ke

kiri dan jarak vertikal sebesar 6 satuan keatas. Untuk itu representasi dari posisi obyek A adalah (-3,6). Sementara obyek B terhadap acuan memiliki jarak horisontal 4 satuan ke kanan dan secara vertikal 3 satuan ke atas sehingga dapat dinotasikan sebagai (4,3).

Sebagai alternatif dari sistem koordinat kartesian, sistem koordinat polar merepresentasikan posisi obyek dalam bentuk vektor dengan dua besaran yaitu jarak antara obyek dengan titik acuan bila ditarik garis lurus, serta sudut antara garis tersebut dengan sumbu horisontal (Strang, 1991). Notasi lokasi pada sistem koordinat polar adalah jarak dan sudut dipisah dengan koma dan dikemas dalam kurung, sebagai contoh (R,θ) . Pada notasi koordinat polar jarak antar titik beserta sudutnya selalu positif dengan posisi sudut berkisar antara 0^0 sampai 360^0 atau 0 hingga 2π untuk representasi radian. Contoh representasi lokasi dengan sistem koordinat polar ditampilkan pada gambar 2.8.



Gambar 2.8 Sistem koordinat polar

Gambar 2.8 menampilkan dua buah obyek A dan B. Obyek A pada gambar memiliki jarak sejauh R_a dari titik acuan dengan posisi sudut θ_a dari sumbu

horizontal positif sehingga notasinya adalah $A(R_a, \theta_a)$. Sementara obyek B memiliki jarak sejauh R_b dari titik acuan dengan sudut θ_b sehingga notasinya $B(R_b, \theta_b)$.

2.2.2 Operasi Vektor

Secara umum, ada dua tipe besaran fisika yaitu skalar dan vektor (Peacock, 2009). Besaran skalar merupakan suatu besaran yang dinotasikan dalam satu ukuran saja. Contoh besaran skalar adalah berat, kelajuan, tinggi, panjang, lebar dan sebagainya. Berbeda dengan besaran skalar, besaran vektor dinotasikan dalam satu nilai ukuran dan satu nilai arah. Sebagai contoh kecepatan sebuah mobil di notasikan sebagai 200km/jam ke arah utara.

Notasi vektor biasa ditampilkan dalam bentuk huruf dengan panah di atasnya namun untuk tujuan penulisan sering kali ditampilkan dalam bentuk huruf bergaris bawah seperti contoh \vec{A} atau \underline{A} (Peacock, 2009). Magnitud atau panjang ukuran dari suatu vektor sering kali dinotasikan sebagai vektor dengan diapit dua tanda pipa, atau hanya hurufnya saja tanpa garis bawah. Sebagai contoh magnitud \underline{A} dinotasikan sebagai $|\underline{A}|$ atau A (Peacock, 2009). Unit vektor merupakan vektor dengan panjang satu satuan, digunakan untuk merepresentasikan sudut dari sebuah vektor dan dinotasikan sebagai huruf dengan sudut menghadap kebawah di atasnya. Sebagai contoh unit vektor dari \underline{A} adalah \hat{A} (Peacock, 2009).

Besaran vektor secara matematis tetap memiliki beberapa sifat dari besaran skalar ketika dioperasikan. Disamping beberapa sifat besaran skalar, vektor juga memiliki karakter tersendiri yang unik dan membedakan dirinya dengan besaran skalar.

Vektor memiliki titik asal yang bebas, selama dua vektor memiliki panjang yang sama dan arah sudut yang sama, maka bisa dikatakan dua vektor tersebut sama (Peacock, 2009). Sebagai contoh pada gambar berikut, titik P dan Q dihubungkan dengan vektor \underline{A} . Titik R dan S memiliki jarak dan posisi sudut yang sama hanya saja bergeser posisinya sehingga vektor antara titik R dan S juga bisa

disebut sebagai vektor A.



Gambar 2.9 Sifat pergeseran vektor (Peacock, 2009)

Penjumlahan vektor merupakan resultan dari dua vektor dengan menempatkan titik awal vektor kedua di titik akhir vektor pertama. Pada penjumlahan vektor berlaku sifat komutatif pada (2.1) (Peacock, 2009).

$$\underline{A} + \underline{B} = \underline{B} + \underline{A} \quad (2.1)$$

Ilustrasinya dapat dilihat pada gambar di bawah. Pada gambar digambarkan bahwa vektor A + B bisa didapatkan dari vektor B yang titik pangkalnya diletakkan pada akhir vektor A maupun sebaliknya.



Gambar 2.10 Sifat komutatif vektor (Peacock, 2009)

Sifat perkalian vektor dengan besaran skalar. Sebuah vektor dapat dikalikan dengan besaran skalar. Hasil perkalian vektor dengan besaran skalar merupakan besaran vektor dengan arah yang sama namun panjangnya beberapa kali lipat sebanyak besaran skalar yang dikalikan dengan vektor tersebut (Peacock, 2009). Dalam perkalian vektor dan besaran skalar, untuk setiap vektor

A dan B, serta besaran skalar α dan β berlaku sifat-sifat pada (2.2), (2.3) dan (2.4).

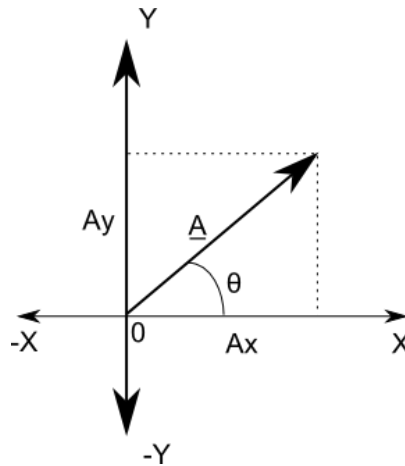
$$\alpha(\underline{A} + \underline{B}) = \alpha\underline{A} + \alpha\underline{B} \quad (2.2)$$

$$(\alpha + \beta)\underline{A} = \alpha\underline{A} + \beta\underline{A} \quad (2.3)$$

$$\alpha(\beta\underline{A}) = (\alpha\beta)\underline{A} \quad (2.4)$$

Perlu diketahui juga bahwa ada beberapa aturan dalam pengoperasian besaran vektor diantaranya sebagai berikut :

- Memungkinkan dilakukan penjumlahan dua buah vektor, namun tidak dimungkinkan dilakukan penjumlahan antara vektor dan besaran skalar.
- Penggunaan dalam persamaan hanya memungkinkan persamaan sejenis yaitu antara vektor dengan vektor atau skalar dengan skalar.
- Tidak ada operasi pembagian dengan pembagi vektor, meskipun bilangan yang dibagi skalar maupun vektor.



Gambar 2.11 Komponen Vektor

Gambar 2.11 menampilkan sebuah vektor dapat diuraikan menjadi dua komponen skalar masing-masing posisi horisontal dan vertikal untuk memudahkan operasi penjumlahan (Favinger, 2008). Penguraian vektor dilakukan dengan menggunakan operasi trigonometri. Untuk sebuah vektor A dapat dinotasikan komponen horisontal A_x dan komponen vertikal A_y . Untuk semua vektor A dengan panjang $|\underline{A}|$ dan sudut θ berlaku (2.5) dan (2.6).

$$A_x = |\underline{A}| \cos \theta \quad (2.5)$$

$$A_y = |\underline{A}| \sin \theta \quad (2.6)$$

Penjumlahan vektor dapat dilakukan dengan menjumlahkan besaran skalar masing-masing komponen vektor yang dijumlahkan (Strang, 1991). Sebagai gambaran misal kita ingin melakukan operasi $\underline{A} + \underline{B}$ dengan asumsi hasilnya adalah \underline{C} . Untuk setiap (2.7) berlaku (2.8) dan (2.9).

$$\underline{A} + \underline{B} = \underline{C} \quad (2.7)$$

$$C_x = A_x + B_x \quad (2.8)$$

$$C_y = A_y + B_y \quad (2.9)$$

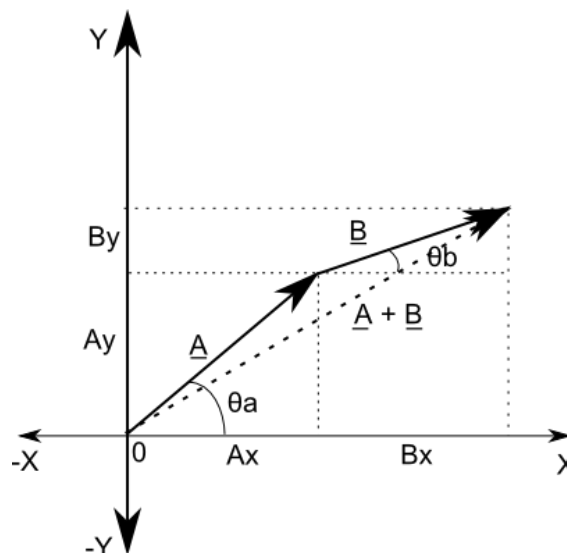
Menggunakan aturan pythagoras berlaku pula (2.10).

$$|C|^2 = C_x^2 + C_y^2 \quad (2.10)$$

Menggunakan aturan trigonometri berlaku (2.11).

$$\theta_c = \tan^{-1} (C_y/C_x) \quad (2.11)$$

Tampilan penjumlahan vektor pada sistem koordinat kartesian ditampilkan pada gambar 2.12.



Gambar 2.12 Penjumlahan vektor

Sebagaimana penjumlahan vektor, pengurangan vektor dapat dilakukan dengan mengurangkan besaran skalar masing-masing komponen vektor yang

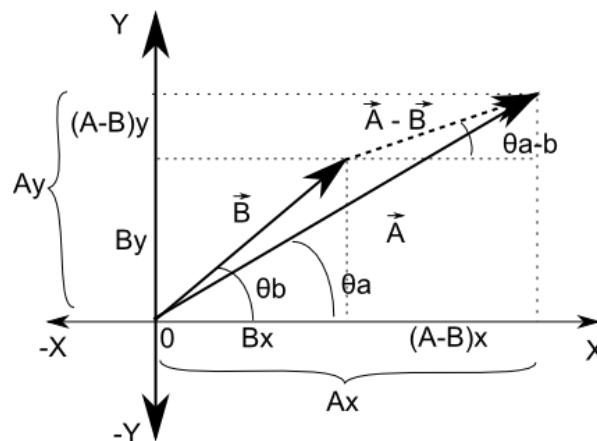
dikurangkan (Strang, 1991). Sebagai gambaran misal kita ingin melakukan operasi $\underline{A} - \underline{B}$ dengan asumsi hasilnya adalah \underline{C} . Untuk setiap (2.12) berlaku (2.13) dan (2.14)

$$\underline{A} - \underline{B} = \underline{C} \quad (2.12)$$

$$C_x = A_x - B_x \quad (2.13)$$

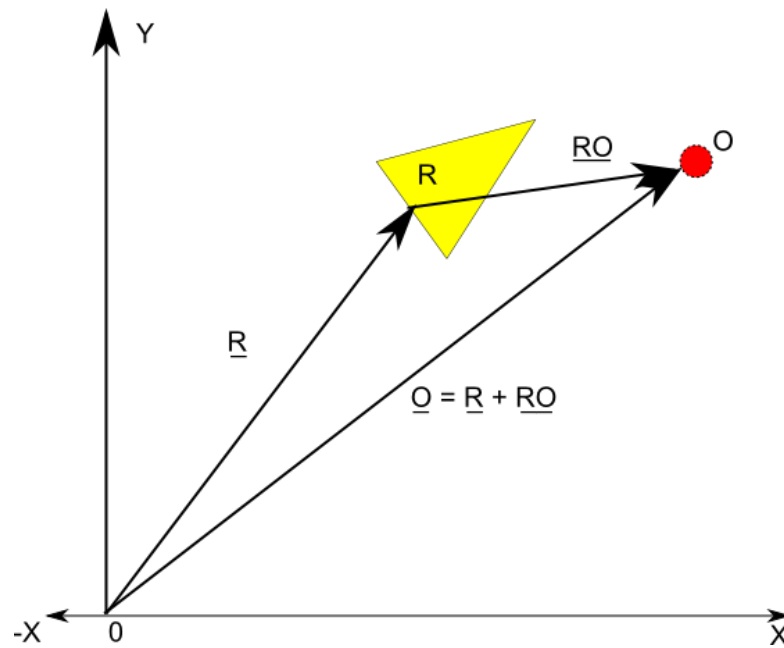
$$C_y = A_y - B_y \quad (2.14)$$

Dengan aturan pythagoras dan trigonometri berlaku pula (2.10) dan (2.11) sebagaimana pada penjumlahan vektor. Tampilan pengurangan vektor pada sistem koordinat kartesian ditampilkan pada gambar 2.13.



Gambar 2.13 Pengurangan Vektor

Penelitian ini banyak menggunakan relatifitas antara obyek satu dengan yang lain. Sebagai contoh posisi suatu obyek relatif terhadap robot, maupun posisi robot relatif terhadap titik acuan. Dengan dideklarasikannya posisi obyek-obyek dan robot dalam notasi koordinat polar, notasi tersebut juga bisa dianggap sebagai vektor. Untuk merepresentasikan posisi obyek relatif terhadap suatu titik acuan bisa menggunakan resultan vektor posisi robot relatif terhadap titik acuan dan posisi obyek relatif terhadap robot. Gambar 2.14 menampilkan posisi obyek terhadap titik acuan yang merupakan resultan vektor posisi robot terhadap titik acuan dengan vektor posisi obyek terhadap robot.



Gambar 2.14 Vektor posisi obyek terhadap titik acuan

2.2.3 Data Warehouse

Data warehouse merupakan sebuah tempat penyimpanan data terpisah dari sistem utama yang menampung informasi-informasi yang dapat membantu proses pengambilan keputusan (Turkmen, 2007). Pada *data warehouse* dapat dimuat informasi-informasi penting dari basis data operasional tanpa mengganggu proses operasional itu sendiri. Dengan demikian proses analisa informasi dapat diambil dengan cepat sesuai kebutuhan. Menurut Turkmen pada laporan thesisnya, *data warehouse* memiliki beberapa karakteristik sebagai berikut.

- Berorientasi subyek. *Data warehouse* dirancang untuk menyesuaikan kebutuhan pengambilan keputusan. Data yang ada pada sistem operasional sering kali berbentuk transaksional sehingga sulit dibaca informasinya, oleh karena itu informasi pada *data warehouse* dikemas sedemikian rupa agar menyesuaikan dengan kebutuhan pengambilan keputusan.
- Terintegrasi. Seluruh informasi pada *data warehouse* harus terintegrasi dengan baik. Format-format tabel, bentuk satuan, hingga penamaan

diatur secara konsisten dan ketat agar tidak saling berbenturan antara data satu dengan yang lain. Pengaturan format ini juga dimaksudkan untuk memudahkan pengambilan informasi di sisi pengguna.

- *Nonvolatile*. Informasi pada *data warehouse* tidak boleh berbeda dari data aslinya. Pada proses pengambilan data hanya boleh dilakukan perubahan format maupun konversi satuan saja. Hal ini dimaksudkan agar seluruh informasi yang ada pada *data warehouse* merupakan informasi yang sebenarnya. Demikian pula proses-proses pada *data warehouse* juga tidak boleh merubah informasi yang ada pada basis data operasionalnya.

- Relatif terhadap waktu. Informasi pada *data warehouse* selain disusun menurut index tertentu juga diatur secara relatif terhadap waktu. Karena data operasional yang terus bergerak, setiap pengambilan data dari basis data operasional harus dilengkapi varian waktu. Varian waktu yang ditambahkan pada data difungsikan untuk menjaga konsistensi serta memfasilitasi bila pada proses pengambilan keputusan diperlukan pembatasan lingkup waktu tertentu.

Membangun *data warehouse* memerlukan beberapa proses dalam mempersiapkan data. Proses tersebut diperlukan untuk menyesuaikan data dari basis data operasional agar sesuai dengan skema pada *data warehouse* sehingga dapat digunakan untuk keperluan analisa (Turkmen, 2007). Beberapa proses yang umum dilakukan dalam mengambil data diantaranya adalah sebagai berikut.

- Ekstraksi data. Tujuan dari ekstraksi data pada intinya adalah mengambil data dari satu tempat ke tempat yang lain sebelum dimodifikasi.

- Konversi data. Proses konversi diperlukan untuk menyesuaikan besaran dan satuan yang digunakan pada basis data operasional dengan besaran dan satuan pada *data warehouse*.

- Pembersihan data. Data operasional sering kali menyimpan informasi-informasi yang telah tidak dipergunakan atau tidak relevan dengan proses pengambilan keputusan.

- Integrasi. Pada *data warehouse* dimungkinkan terjadi pengambilan informasi dari beberapa sumber sekaligus, hal ini memerlukan proses integrasi antara data yang berbeda-beda.
- Transformasi. Proses transformasi data diperlukan untuk menyesuaikan informasi yang ada dengan skema pada *data warehouse*.

Arsitektur *data warehouse* memiliki berbagai bentuk dengan karakteristik masing-masing. Berbagai bentuk *data warehouse* didasarkan pada kebutuhan informasi untuk dianalisa. Beberapa contoh arsitektur *data warehouse* diilustrasikan dalam gambar 2.15.

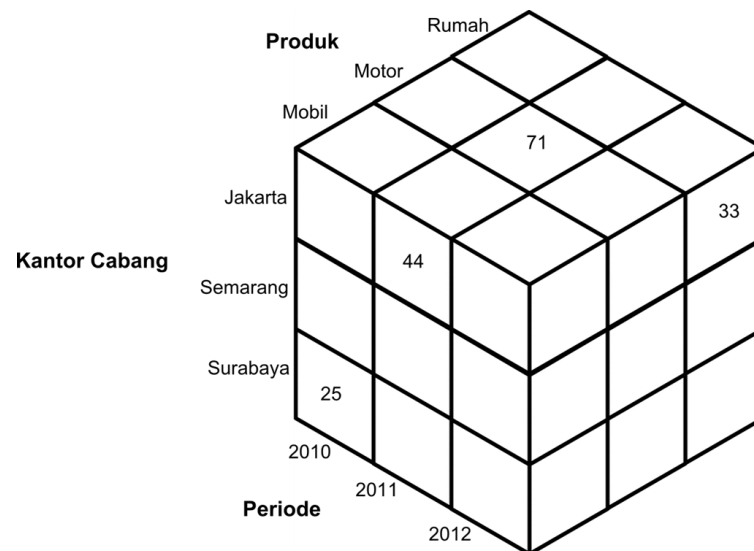


Gambar 2.15 Beberapa tipe arsitektur data warehouse (Sen dan Sinha, 2005)

2.2.4 OLAP

Data warehouse hanyalah sekumpulan data yang dikelola secara terintegrasi. Untuk kebutuhan pengambilan keputusan masih diperlukan lagi alat bantu yang menjembatani antara pengguna dengan *data warehouse*. Salah satu alat bantu dalam pengambilan keputusan adalah OLAP. OLAP pertama kali

diusulkan oleh E.F Codd yang merepresentasikan informasi ke dalam larik multidimensi (Hammergren dan Simon, 2009). Contoh struktur sebuah OLAP 3 dimensi diilustrasikan pada gambar 2.16.



Gambar 2.16 OLAP 3 dimensi

Dalam melakukan proses analisa, diperlukan sebuah ukuran numerik yang mewakili informasi yang dianalisa. Sering kali pada proses kalkulasi untuk mendapatkan angka tersebut memerlukan berbagai dimensi variabel yang tersedia pada *data warehouse*. Sesuai kebutuhan variabel-variabel tersebut, OLAP terdiri dari beberapa komponen (Bimonte dan Tchounikine, 2007) diantaranya adalah sebagai berikut :

- Dimensi. Merupakan suatu dimensi variabel yang merepresentasikan pembagian informasi berdasarkan kelompok atau kategori tertentu.
- Anggota. Merupakan elemen-elemen dari sebuah dimensi.
- Hirarki. Merupakan struktur sebuah dimensi dari yang paling global hingga ke anggota yang lebih spesifik
- Fakta. Merupakan informasi yang tersimpan pada satu irisan antara dimensi yang satu dengan yang lain.

Pada gambar 2.16 yang menjadi dimensi adalah kantor cabang, produk dan periode. Kantor cabang memiliki anggota Jakarta, Semarang dan Surabaya. Pada contoh lain misal kantor cabangnya lebih dari tiga misalnya Jakarta, Bandung Semarang, Solo, Surabaya dan Malang, dapat dikelompokkan kedalam hirarki tertentu misalnya Jawa Tengah beranggotakan Semarang dan Solo sementara Jawa Timur beranggotakan Surabaya dan Malang. Fakta dihimpun dari data yang ada pada *data warehouse* dengan batasan dimensi dimana fakta tersebut berada. Sebagai contoh fakta penjualan mobil di Surabaya pada 2010 sejumlah 25 unit, penjualan rumah di Jakarta pada 2012 sejumlah 33 unit dan seterusnya.

Dalam memanfaatkan informasi pada OLAP, berlaku beberapa operasi. Operasi OLAP difungsikan untuk memandangi informasi pada OLAP dari berbagai sudut pandang berbeda (Hammergren dan Simon, 2009). Beberapa operasi yang didukung oleh OLAP multi dimensi diantaranya sebagai berikut.

- **Konsolidasi.** Merupakan pengelompokan beberapa kategori sesuai hirarkinya untuk memperoleh informasi yang lebih global. Misal pada contoh OLAP sebelumnya, fakta-fakta dari beberapa kota dalam satu provinsi di konsolidasikan untuk melakukan analisa yang lebih luas.
- **Perincian.** Merupakan kebalikan dari konsolidasi, operasi ini memperinci satu kategori pada dimensi menjadi anggota-anggota yang lebih kecil lingkupnya. Contoh pada OLAP sebelumnya penjualan mobil dirinci lagi dengan pengelompokan berdasarkan merk.
- **Pencacahan.** Merupakan operasi membandingkan fakta-fakta yang ada dalam satu lingkup dimensi saja. Sering kali diperlukan untuk mencari hubungan antar dimensi yang dianalisa. Sebagai contoh pada OLAP sebelumnya dilihat data penjualan mobil saja untuk semua kota dari tahun ke tahun bagaimana trennya.

BAB III

METODE PENELITIAN

3.1 Bahan Penelitian

Bahan-bahan penelitian ini meliputi beberapa hal sebagai berikut:

- Komponen robotik meliputi 2 unit motor *stepper*, 1 unit motor *servo*, 1 unit sensor sonar, 2 unit *stepper driver*, 1 unit modul komunikasi dan 1 set *microcontroller*.
- *Programmer microcontroller* yang digunakan lengkap dengan *library-library* dasar untuk pengembangan lebih lanjut.
- *Installer* server untuk data warehouse meliputi web server dan database server berbasis open source.

3.2 Alat Penelitian

Penelitian kali ini menggunakan peralatan-peralatan sebagai berikut :

- Sebuah laptop sebagai pusat dokumentasi penelitian sehingga praktis dan dapat dipergunakan secara dinamis.
- Alat ukur metrik berupa mistar dan busur untuk kalibrasi selisih dan toleransi yang ada antara kondisi nyata di lapangan dengan informasi yang harus diolah pada *data warehouse*.
- Aplikasi manajemen basis data untuk memantau posisi dan isi data sesaat secara manual.

3.3 Prosedur Penelitian

Sebagai titik awal penelitian, peneliti mengambil beberapa asumsi sebagai berikut :

- Posisi dan jarak antar obyek acak dan tidak teratur.
- Bentuk dan ukuran obyek sama yaitu silinder.
- Agen yang digunakan adalah sistem robot sederhana tanpa sistem navigasi sehingga tidak dapat diukur kemana robot mengarah dan pada posisi koordinat berapa robot berada.
- Untuk setiap percobaan, posisi awal agen acak antara satu dengan yang lain.
- Sensor dan aktuator pada penelitian ini dianggap ideal, sehingga tidak banyak

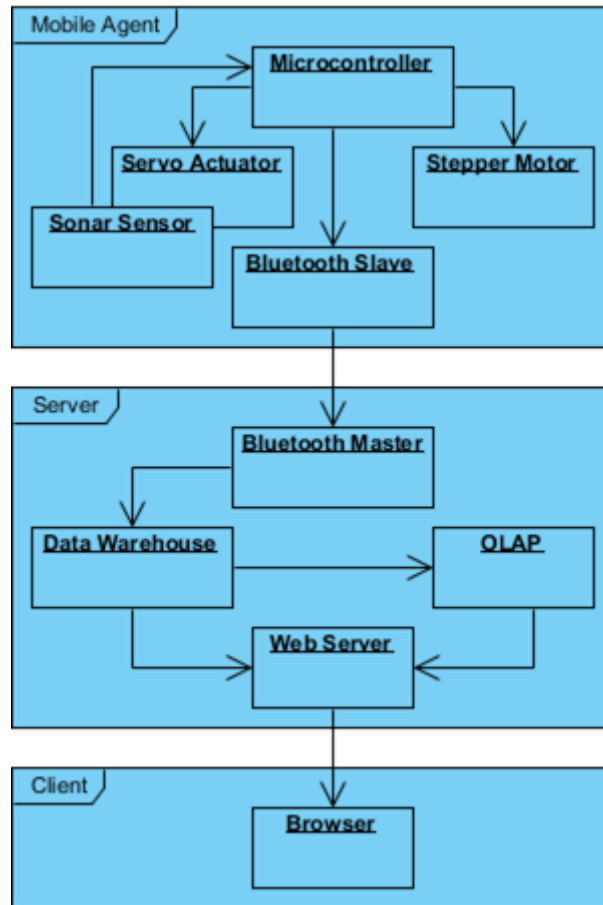
membahas metode-metode filter deteksi obyek.

3.4 Perancangan Blok Sistem

Secara garis besar sistem yang dibuat melibatkan tiga elemen utama yaitu agen, *server* dan *client*. Agen pada penelitian ini merupakan sebuah robot sederhana dengan berbagai komponen diantaranya adalah *microcontroller*, servo, sonar, motor *stepper* dan modul komunikasi. Microcontroller berfungsi sebagai pengendali utama robot yang mengatur seluruh komponen robot. Motor servo mengatur posisi arah sensor sonar. Motor *stepper* sebagai aktuator utama menggerakkan robot ke arah yang diinginkan. Modul komunikasi yang digunakan adalah *bluetooth* dengan konfigurasi sebagai *slave*.

Server pada penelitian ini memiliki beberapa komponen diantaranya modul komunikasi, *data warehouse*, olap dan *web server*. Modul komunikasi yang digunakan adalah bluetooth dengan konfigurasi sebagai master sehingga dapat berinteraksi dengan agen. Komponen selanjutnya adalah *data warehouse* dan OLAP yang akan dirinci perancangannya pada bagian selanjutnya.

Client pada penelitian ini berupa peramban yang digunakan pengguna mengakses sistem. Sistem ini dapat diakses menggunakan berbagai macam peramban, namun penulis menyarankan menggunakan Mozilla Firefox atau Google Chrome. Rancangan blok sistem beserta alur proses secara keseluruhan ditampilkan pada gambar 3.1.

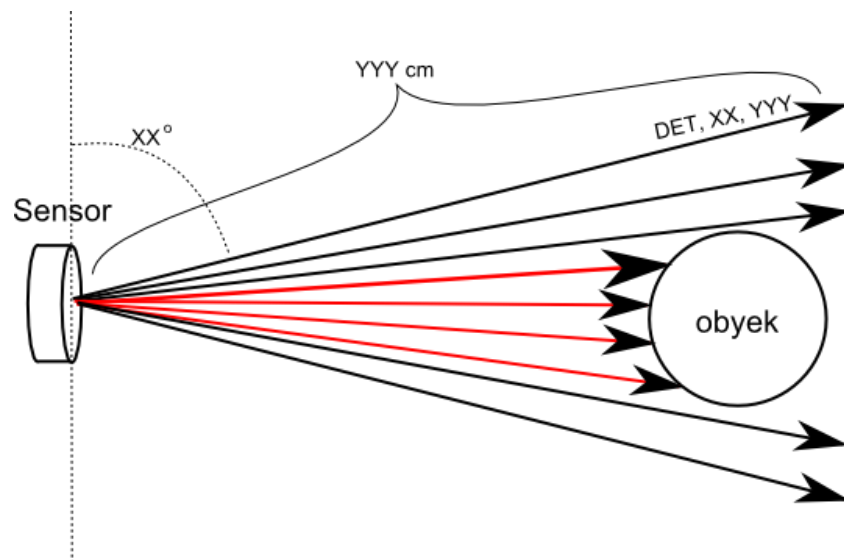


Gambar 3.1 Diagram blok sistem

3.5 Sistem Robotik

Dengan berbagai asumsi sebelumnya, penelitian dapat dimulai dengan menentukan besaran-besaran koefisien. Tahapan ini meliputi kalibrasi mekanik robot serta penentuan besaran-besaran koefisien antara kondisi fisik dengan data sebagai pembanding sehingga informasi disimpan secara akurat. Besaran-besaran ini ditentukan dengan melakukan percobaan deteksi obyek di berbagai posisi dan membandingkan dengan posisi sebenarnya berdasarkan pengukuran fisik.

Proses yang pertama kali dilakukan adalah menentukan profil deteksi obyek. Profil deteksi merupakan sekumpulan deteksi obyek dari berbagai sudut yang menginformasikan keberadaan sebuah obyek di suatu sudut dan jarak tertentu. Dalam menentukan profilnya dilakukan serangkaian percobaan dengan menggunakan robot yang ada. Metode pendeteksiannya diilustrasikan dalam gambar 3.2.



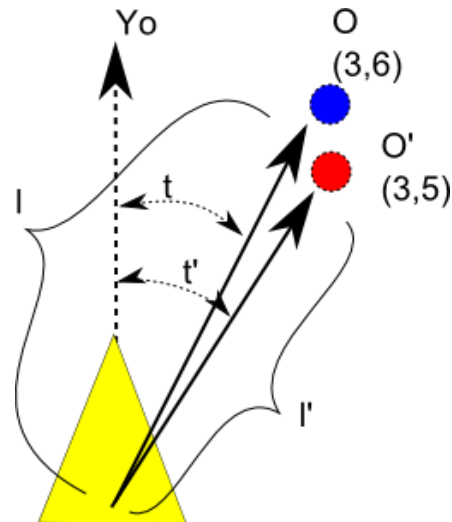
Gambar 3.2 Profil deteksi obyek

Gambar 3.2 di atas menampilkan profil deteksi obyek pada posisi tertentu terhadap sensor. Dengan asumsi obyeknya berupa silinder, tampilan penampang melintang sebuah obyek dapat digambarkan sebagai lingkaran. Pada proses deteksi, sonar mengirimkan sinyal-sinyal ultrasonik kemudian mendeteksi adanya pantulan sinyal tersebut. Pada posisi sudut dimana tidak ada obyek, sinyal hanya lewat dan tidak dipantulkan pada obyek sehingga tidak ditemukan deteksi seperti ditampilkan pada garis warna hitam. Sementara ketika ada obyek, sinyal dipantulkan kembali sehingga dideteksi oleh sensor seperti ditampilkan pada garis warna merah. Dengan asumsi penampang obyek berbentuk lingkaran dan sudut antara deteksi yang satu dengan yang lain sangat kecil, titik tengah obyek bisa ditentukan dengan jarak deteksi yang paling dekat. Meskipun demikian tingkat ketelitian sensor yang terbatas terkadang tidak menghasilkan perbedaan signifikan sehingga sulit dicari titik terdekatnya. Untuk mengatasi hal ini, digunakan alternatif sudut median dari seluruh deteksi yang dihasilkan.

Sebagaimana dibahas pada tinjauan pustaka, pendeteksian menggunakan sonar sangat rentan terhadap *noise*. Deteksi satu posisi sudut sangat dipengaruhi oleh obyek-obyek di sekitar arah deteksi tersebut hingga mencapai 30^0 (Saleem, 2013). Untuk itu diperlukan metode penanganan yang relevan dengan memprofilkan deteksi obyek tersebut.

Hasil dari profil deteksi pada bagian sebelumnya memberikan dua informasi yaitu sudut deteksi serta jarak obyek dari sensor. Hasil deteksi ini memungkinkan terjadinya simpangan baik sudut maupun jarak dikarenakan struktur sensor yang memposisikan *transmitter* dan *receiver* di posisi terpisah. Untuk menangani simpangan ini dilakukan

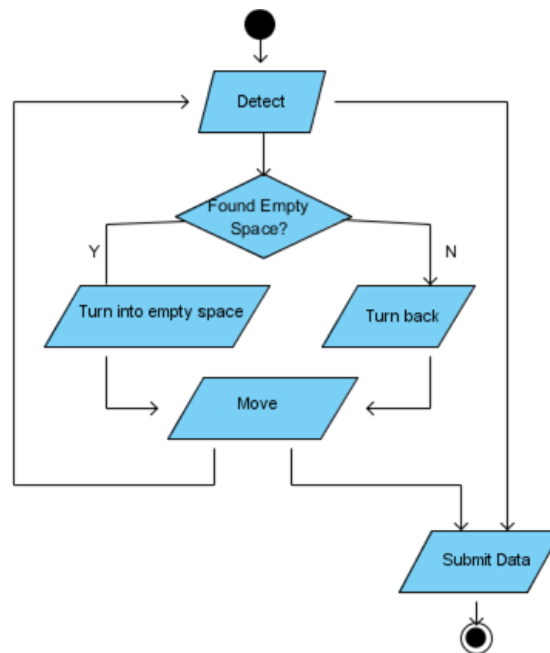
kalibrasi terhadap hasil deteksi obyek dengan melakukan pengukuran fisik sebagai data pembandingnya. Contoh perbedaan hasil deteksi dan pengukuran ditampilkan pada gambar 3.3.



Gambar 3.3 Kalibrasi sensor hasil deteksi dan hasil pengukuran

Gambar 3.3 di atas menampilkan salah satu kemungkinan perbedaan posisi obyek hasil deteksi dan hasil pengukuran. Perbedaan ini didata dari beberapa percobaan serta diperkirakan relasinya apakah linear atau eksponensial serta dibuatkan besaran-besaran perbandingan untuk kemudian digunakan pada percobaan.

Sistem pada penelitian ini menggunakan arduino dengan *microcontroller* berbasis Atmel Atmega. *Controller* dirancang untuk dapat mengendalikan aktuator dan sensor yang dipergunakan. Alur proses pada robot adalah deteksi obyek dengan menggerakkan servo pada sudut-sudut tertentu sambil melakukan deteksi sonar. Data deteksi di setiap sudut dianalisa dan dikirimkan ke sistem. Hasil analisa data deteksi di setiap sudut adalah ada tidaknya ruang kosong, bila ada agen akan mengarah ke ruang kosong tersebut dan bergerak maju, namun bila tidak ada maka agen akan berputar kembali. Data gerakan agen juga dikirimkan ke sistem untuk dianalisa kemudian. Alur proses pada robot ditampilkan pada gambar 3.4.



Gambar 3.4 Flow chart robotik

Antarmuka sistem dengan robot direncanakan untuk dapat mengakomodasi data telemetri dari robot ke sistem serta instruksi dari sistem ke robot. Untuk keleluasaan pergerakan robot, sistem komunikasi antara robot dan sistem dirancang menggunakan telekomunikasi tanpa kabel, sedangkan untuk memenuhi kebutuhan telemetri tipe komunikasi yang digunakan adalah *duplex*. Dua karakteristik komunikasi yang diperlukan menjadi dasar peneliti untuk memilih modul *bluetooth* sebagai media komunikasi.

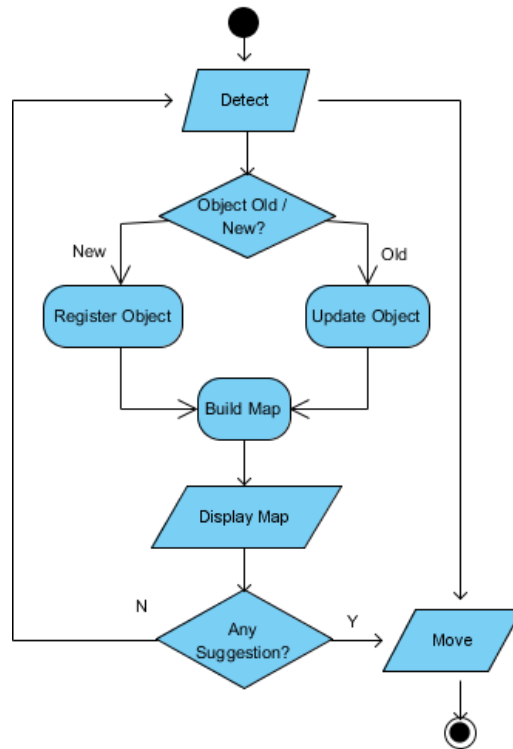
Modul *bluetooth* tipe HC-05 memungkinkan komunikasi serial dilakukan melalui transmisi radio. Dengan menggunakan modul ini, komunikasi dua arah dapat dilakukan secara nirkabel. Sistem dapat menerima hasil deteksi dari robot dan dapat pula mengirim perintah gerakan ke robot, disamping itu proses ini dapat dilakukan secara nirkabel sehingga robot leluasa untuk bergerak.

3.6 Rancang Bangun *Data Warehouse* dan OLAP

Tahapan selanjutnya adalah rancang bangun data warehouse beserta OLAP sebagai pusat informasi SLAM. Pada tahapan ini *data warehouse* mulai dirancang skemanya serta struktur data SLAM yang disimpan pada database. Komposisi data dibentuk seragam sehingga memudahkan proses penggalian informasi SLAM oleh OLAP. Perangkat OLAP disusun untuk menampilkan informasi-informasi yang dibutuhkan. Termasuk pada tahapan ini adalah pengaturan alur data dari hulu ke hilir beserta formula-formula untuk komposisi data menjadi informasi SLAM.

Untuk memudahkan tahapan ini, data dan informasi dikelompokkan dalam beberapa bagian pusat data dengan karakteristik dan fungsional masing-masing. Beberapa kelompok pusat data tersebut antara lain sebagai berikut :

- *Input buffer SLAM*, berupa basis data sementara dimana seluruh data stream dari agen disimpan sebelum dimuat ke dalam basis data. Bersifat transaksional dimana seluruh stream disimpan apa adanya tanpa perubahan.
- *Data warehouse SLAM*, merupakan area dimana seluruh data disimpan dalam format yang seragam untuk meningkatkan keterbacaan data di dalamnya. Pembacaan informasi dari *input buffer SLAM* dilakukan dengan memodifikasi komposisi datanya agar sesuai dengan ragam yang ada pada *data warehouse*. Data pada *input buffer* yang telah terbaca di-flag agar tidak terbaca ulang.
- *OLAP SLAM*, merupakan area yang tempat seluruh informasi penting dihimpun. Informasi pada OLAP berupa tabel fakta secara global yang merujuk pada tabel-tabel lain yang lebih spesifik. Data OLAP berlapis secara historis untuk setiap percobaan dengan menggunakan agen.
- *Output buffer SLAM*, berupa basis data sementara yang mengalokasikan informasi-informasi spesifik yang dibutuhkan sistem untuk menampilkan peta global. *Output buffer* memiliki ukuran data yang dinamis sesuai jumlah obyek yang digunakan pada percobaan. Informasi yang disimpan pada *output buffer* sifatnya terkini dan bila ada perubahan informasi akan memperbaharui informasi lama pada posisi yang sama tanpa membuat rekaman baru.

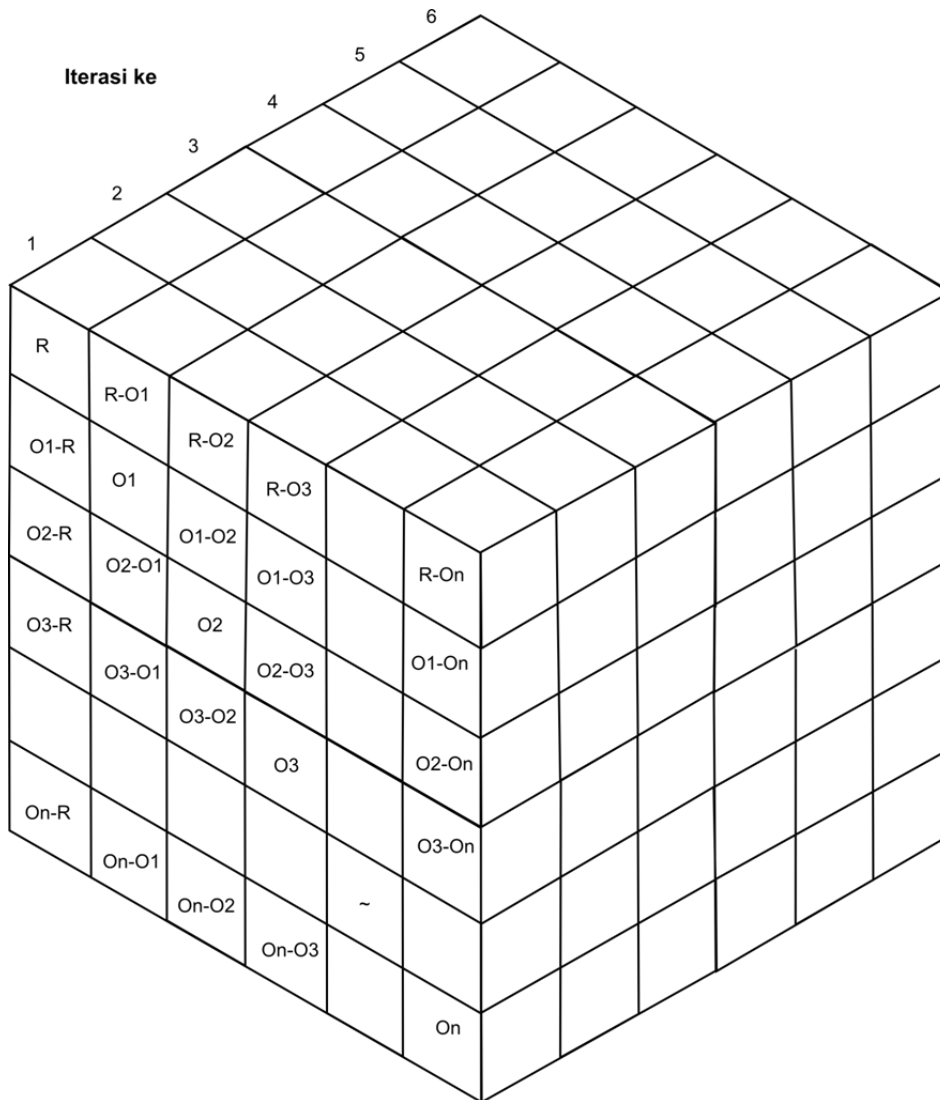


Gambar 3.5 Flow chart sistem

Gambar 3.5 menampilkan proses pertukaran dan pengolahan data dan informasi terjadi pada *buffer*, *data warehouse* dan OLAP sistem SLAM. Berbagai perhitungan matematis diperlukan agar memperoleh informasi yang relevan sehingga menghasilkan sistem yang dapat diandalkan.

Sebagai solusi visualisasi SLAM, disusun sebuah skema OLAP. Skema OLAP yang dibuat menampung informasi-informasi yang diperlukan untuk proses visualisasi pada tahapan selanjutnya. Informasi yang dimuat pada skema ini diantaranya adalah sebagai berikut.

- Posisi agen dan obyek relatif terhadap titik acuan
- Relasi posisi antara agen dan obyek serta antar obyek satu dengan yang lain



Gambar 3.6 Skema OLAP SLAM

Gambar 3.6 di atas menampilkan skema OLAP yang digunakan dalam penelitian ini. Skema OLAP di atas menampilkan informasi yang sama pada dua dimensi utamanya hanya saja posisinya saling berlawanan. Seperti tampak pada gambar, pada sumbu diagonal dicantumkan fakta-fakta posisi robot dan obyek-obyek yang ada pada sistem sementara di kolom dan baris lain ditampilkan relasi antara dua obyek atau antara robot dan salah satu obyek. Dimensi ketiga dari skema OLAP yang diusulkan adalah dimensi waktu atau posisi step iterasi selama proses penelitian. Fakta-fakta yang ditampilkan pada OLAP disimpan dalam bentuk sistem koordinat polar berupa vektor-vektor.

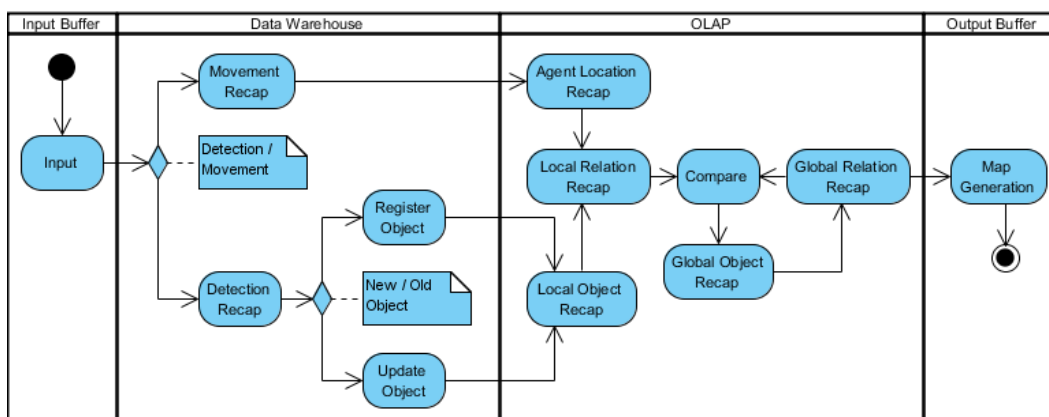
Sebelum melakukan rancang bangun lebih lanjut, dilakukan analisa dan perancangan sistem. Bagian ini menampilkan permodelan sistem informasi dalam bentuk diagram-diagram agar mudah dipahami alur proses dan fungsional tiap-tiap komponennya. Sistem informasi dirancang dilakukan menggunakan konsep perancangan berorientasi obyek dengan

merepresentasikan sistem informasi kedalam diagram-diagram UML. Perancangan sistem informasi pada penelitian ini disusun dalam tiga tahapan diantaranya adalah *functional modelling*, *structural modelling* dan *behavioral modelling*.

3.6.1 Functional Modelling

Perancangan pertama pada penelitian ini adalah *functional modelling*. Bagian ini menampilkan permodelan sistem secara fungsional. Sistem informasi digambarkan secara mendasar mengenai fungsi-fungsi apa saja yang harus dapat dilakukan oleh sistem. Diagram-diagram yang digambarkan pada *functional modelling* adalah *activity diagram* dan *use case diagram*.

Pada *activity diagram* digambarkan keseluruhan aktifitas yang terjadi pada sistem secara global. Aktifitas-aktifitas yang digambarkan dikelompokkan pada elemen-elemen yang melakukan aktifitas tersebut. Pada penelitian ini *activity diagram* ditampilkan pada gambar 3.7.



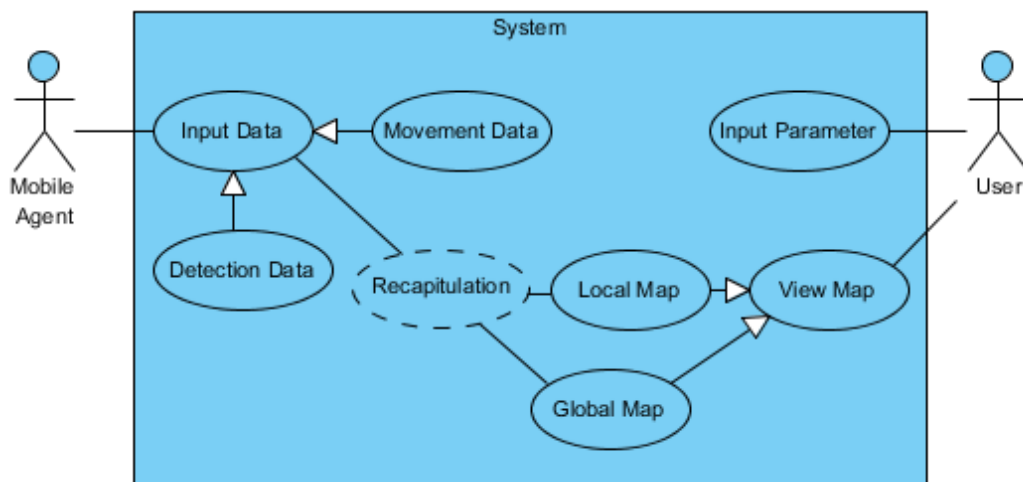
Gambar 3.7 Activity diagram OLAP SLAM

Pada gambar 3.7 di atas ditampilkan bahwa sistem dibagi menjadi empat bagian yaitu *input buffer*, *data warehouse*, OLAP dan *output buffer*. Proses dimulai dari input data pada komponen *input buffer*. Data yang diterima sistem kemudian disaring apakah data tersebut merupakan data gerakan maupun deteksi. Data gerakan distandarkan formatnya pada *data warehouse* untuk kemudian dilakukan rekapitulasi lokasi agen oleh OLAP. Data deteksi direkapitulasi posisinya untuk kemudian dianalisa apakah obyek tersebut sudah ada atau belum pada peta lokal. Bila obyek sudah ada pada peta, dilakukan update posisi sementara bila belum ada akan ditambahkan pada peta lokal dan terakhir keseluruhan posisi obyek ini direkapitulasi posisinya pada OLAP. Posisi obyek-obyek pada peta lokal dilakukan rekapitulasi relasi antar obyek sehingga keseluruhan informasi yang ada dapat disusun

menjadi sebuah peta lokal.

Peta global merupakan komponen yang menyimpan seluruh informasi hasil observasi. Bila peta global belum terisi, besaran-besaran pada peta lokal pertama menjadi titik acuan awal peta global. Peta lokal selanjutnya dibandingkan dengan isi dari peta global sehingga ditemui relasi antara peta global dengan peta lokal yang sedang diproses observasinya. Relasi antara peta global dengan peta lokal menjadi dasar rujukan obyek-obyek pada peta lokal tersebut untuk dipetakan pada peta global.

Diagram kedua yang ditampilkan pada *functional modelling* adalah *use case diagram* yang menampilkan fungsi-fungsi sistem dari sudut pandang pengguna. *Use case diagram* menampilkan sistem sebagai satu kesatuan dengan aktor-aktor berada di sekitarnya melakukan interaksi terhadap sistem. Hasil penelitian ini menampilkan *use case diagram* seperti gambar 3.8.



Gambar 3.8 Use Case diagram OLAP SLAM

Use case diagram pada gambar 3.8 di atas menampilkan ilustrasi yang lebih sederhana tentang sistem. Sebagaimana pada gambar di atas ditampilkan dua aktor utama yang berperan dalam sistem ini yaitu agen dan pengguna dari sistem. Hal yang dilakukan agen pada dasarnya hanyalah input data dimana data yang diinput dapat dijabarkan sebagai data deteksi dan data gerakan. Pengguna sistem hanya melakukan parameter awal dan dapat menampilkan peta dimana peta tersebut bisa berupa peta lokal maupun global. Peta yang dihasilkan oleh sistem merupakan hasil rekapitulasi dari data yang sudah masuk.

3.6.2 Structural Modelling

Perancangan kedua adalah permodelan sistem secara struktural yang menampilkan

Proses penyusunan peta ini memerlukan adanya persiapan pemetaan baik di sisi peta lokal maupun global.

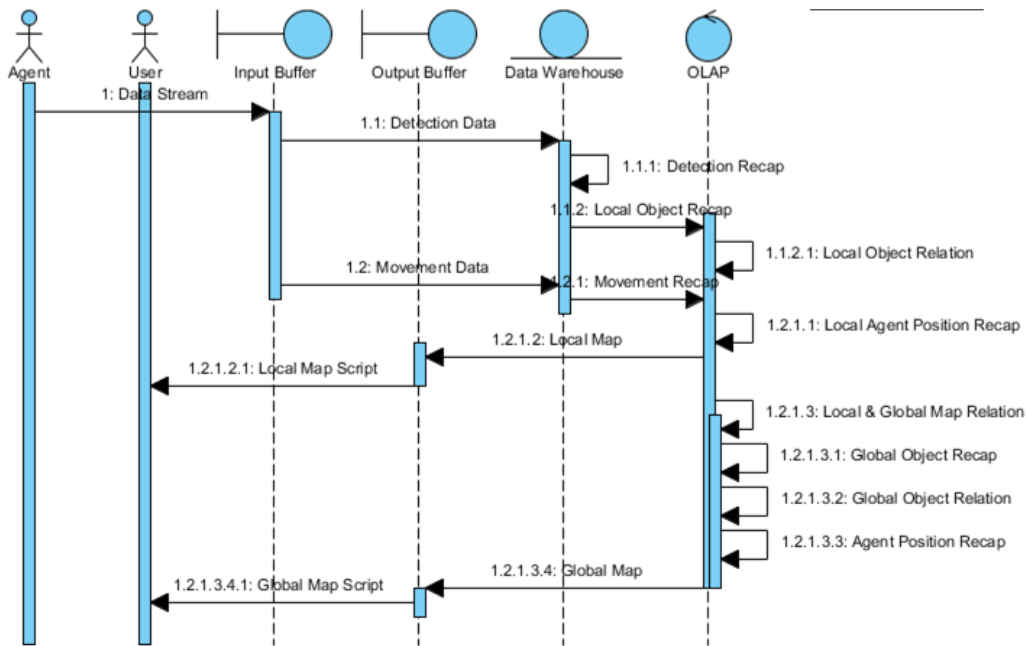
Peta Global merupakan komposisi dari posisi agen, rekapitulasi posisi obyek serta relasi posisi obyek pada peta global. Relasi antar obyek direkapitulasi dari posisi obyek itu sendiri yang merupakan hasil pemetaan obyek-obyek pada peta lokal. Pemetaan obyek lokal ke peta global serta rekapitulasi posisi agen pada peta global didapatkan dari informasi relasi antara peta lokal dan peta global. Relasi antara peta lokal dan peta global merupakan hasil dari perbandingan relasi antar obyek pada peta lokal dan peta global.

Informasi yang tersedia pada peta lokal merupakan komposisi dari rekapitulasi gerakan agen, rekapitulasi posisi obyek serta relasi antar obyek pada peta lokal. Relasi antar obyek merupakan hasil dari perbandingan posisi antar obyek pada peta lokal yang merupakan rekapitulasi dari deteksi agen pada area observasi. Rekapitulasi gerakan agen dan rekapitulasi deteksi didapatkan dari daftar deteksi dan gerakan yang merupakan hasil penyaringan informasi yang diterima oleh sistem melalui *input buffer*.

3.6.3 Behavioral Modelling

Permodelan terakhir pada perancangan sistem ini adalah *behavioral modelling*. Pada permodelan ini dirancang alur proses dan perilaku setiap komponen sistem yang dibuat. Diagram yang digunakan untuk menggambarkan sistem pada permodelan ini adalah *sequence diagram*.

Sequence diagram menampilkan aktor-aktor beserta entitas-entitas yang terlibat dalam sistem. Setiap aktor dan entitas yang ada memiliki garis vertikal dimana proses terjadi yang disebut sebagai *lifeline*. Tingkah laku dari aktor dan entitas ditampilkan sebagai alur pergerakan informasi yang digambarkan sebagai anak panah yang menghubungkan *lifeline* yang terlibat pada alur informasi tersebut. Hasil permodelan menggunakan *sequence diagram* ditampilkan pada gambar 3.10.



Gambar 3.10 Sequence diagram OLAP SLAM

Gambar 3.10 di atas menampilkan dua aktor dan empat entitas yang terlibat sepanjang pemrosesan pada sistem. Alur proses dimulai dengan pengiriman data dari agen ke *input buffer*. Pada *input buffer* alur informasi dibagi dua menjadi data deteksi dan data gerakan yang kemudian disimpan pada *data warehouse*. Pada *data warehouse* data deteksi direkapitulasi kemudian dianalisa hingga menghasilkan recap posisi obyek pada OLAP. Posisi obyek pada OLAP dibandingkan satu sama lain untuk menghasilkan relasi antar obyek. Data gerakan yang diterima oleh *data warehouse* juga dianalisa oleh OLAP untuk merekapitulasi posisi agen pada peta lokal. Data recap posisi dan relasi obyek serta posisi agen diproses oleh OLAP dan dikirimkan sebagai informasi peta lokal pada *output buffer* yang kemudian diakses oleh pengguna.

Seluruh informasi peta lokal pada OLAP kemudian direkapitulasi untuk menemukan relasi antar peta lokal dengan peta global. Setelah relasi antara peta lokal dan peta global didapatkan, posisi obyek, relasi antar obyek serta posisi agen pada peta lokal dipetakan ke dalam peta global. Seluruh informasi hasil pemetaan pada peta global disusun untuk dikirimkan sebagai informasi peta global pada *output buffer* untuk kemudian diakses oleh user.

3.6.4 Rekapitulasi Gerakan Agen pada Peta Lokal

Proses rancang bangun selanjutnya adalah mengembangkan sistem rekapitulasi gerakan agen. Rincian gerakan robot diterima oleh sistem dari *buffer* dengan melalui

beberapa proses pengolahan data diantaranya sebagai berikut.

- Penyaringan tipe data. Yang menghasilkan informasi rekapitulasi gerakan agen hanyalah data gerakan sehingga dilakukan penyaringan tipe data dari *buffer* yaitu data yang memiliki kode MOV.
- Konversi besaran dan satuan. Besaran dan satuan yang digunakan pada *buffer* masih berupa jumlah step. Besaran input dari *buffer* dikonversi kedalam satuan milimeter untuk gerakan maju dan satuan derajat untuk gerakan memutar baik ke kanan maupun ke kiri. Proses konversi menggunakan koefisien yang didapatkan pada proses kalibrasi dan penentuan besaran.

Setelah melalui proses penyaringan dan konversi satuan, data dari *buffer* disimpan pada *data warehouse*. Tujuan proses konversi dan penyaringan ini adalah untuk menghasilkan tipe data yang seragam sesuai struktur tabel pada *data warehouse*. *Data warehouse* memberikan informasi tambahan yang merupakan penanda waktu ketika data diambil dari *buffer*. Informasi waktu pengambilan data disimpan sebagai antisipasi bila ada kebutuhan di kemudian hari.

Posisi terakhir agen pada peta lokal dinotasikan dalam dua vektor yaitu vektor $\overrightarrow{A_{mn}}$ yang menyimpan kedudukan polar agen terhadap titik acuan peta lokal serta unit vektor $\overrightarrow{B_{mn}}$ yang menyimpan sudut posisi arah agen terhadap titik acuan peta lokal. Gerakan memutar baik ke kanan maupun ke kiri pada penelitian ini dianggap sebagai persiapan untuk pergerakan robot sehingga proses rekapitulasi diperhitungkan setiap kali terjadi gerakan maju. Perhitungan vektor pada proses rekapitulasi ini adalah sebagai berikut.

Untuk setiap $\overrightarrow{A_i}$ merupakan vektor gerakan robot ke arah sudut α dan maju sejauh l berlaku (3.1).

$$\overrightarrow{A_{mn}} = \sum \overrightarrow{A_i} \quad (3.1)$$

Untuk l_{mn} merupakan panjang dari vektor $\overrightarrow{A_{mn}}$ dan α_{mn} merupakan sudut dari vektor $\overrightarrow{A_{mn}}$ dengan l_i dan α_i merupakan jarak dan arah gerakan, sesuai aturan penjumlahan vektor berlaku (3.2) dan (3.3).

$$\begin{aligned} l_{mn} &= \sqrt{(\sum l_{ix})^2 + (\sum l_{iy})^2} \\ l_{mn} &= \sqrt{(\sum l_i \cos \alpha_i)^2 + (\sum l_i \sin \alpha_i)^2} \\ \alpha_{mn} &= \tan^{-1} \frac{(\sum l_{iy})}{(\sum l_{ix})} \end{aligned} \quad (3.2)$$

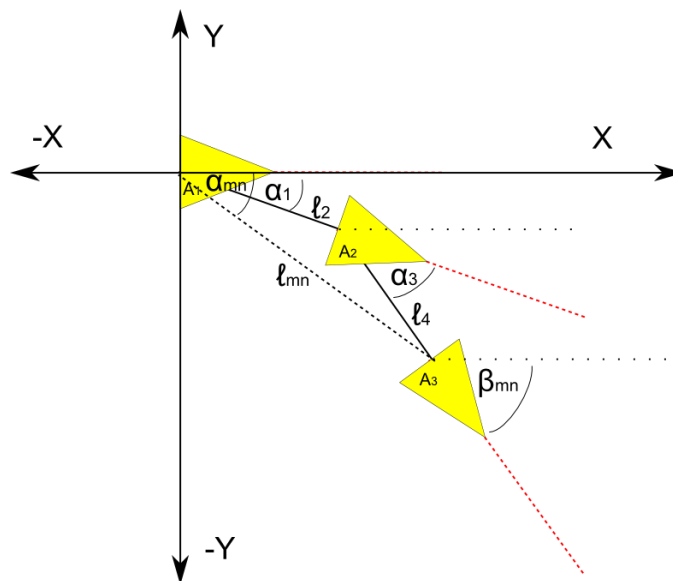
$$\alpha_{mn} = \tan^{-1} \frac{(\sum l_i \sin \alpha_i)}{(\sum l_i \cos \alpha_i)} \quad (3.3)$$

Untuk β_{mn} merupakan sudut dari unit vektor $\overrightarrow{B_{mn}}$ berlaku (3.4).

$$\beta_{mn} = \sum \alpha_i \quad (3.4)$$

Persamaan (3.2), (3.3) dan (3.4) di atas menghasilkan dua besaran vektor, namun dikarenakan besaran vektor kedua merupakan unit vektor sehingga dapat disimpan dalam tiga kolom variabel yaitu satu besaran skalar jarak dan dua besaran skalar sudut.

Gambar 3.11 menampilkan agen sebagai segitiga kuning yang mula-mula berada pada posisi A1. Posisi A1 ini juga menjadi acuan awal peta lokal dengan titik O koordinat terletak pada pangkal agen dan sumbu X positif merupakan arah agen menghadap. Pergerakan agen dimulai dengan arah sudut α_1 kemudian maju sejauh l_2 dan seterusnya. Posisi akhir agen ditampilkan sejauh l_{mn} dari titik O pada sudut α_{mn} dan posisi agen menghadap ke arah β_{mn} . Acuan besaran sudut adalah positif untuk perputaran ke kiri dan negatif untuk perputaran ke kanan.



Gambar 3.11 Rekapitulasi gerakan agen

3.6.5 Rekapitulasi Posisi Obyek pada Peta Lokal

Proses rancang bangun selanjutnya mengembangkan sistem rekapitulasi posisi obyek. Sebagaimana pada rekapitulasi gerakan, rekapitulasi posisi obyek didapatkan sistem dari *buffer* dengan melalui beberapa proses pengolahan data diantaranya sebagai berikut.

- Penyaringan tipe data. Yang menghasilkan informasi rekapitulasi posisi obyek lokal hanyalah data deteksi sehingga dilakukan penyaringan tipe data dari *buffer* yaitu data

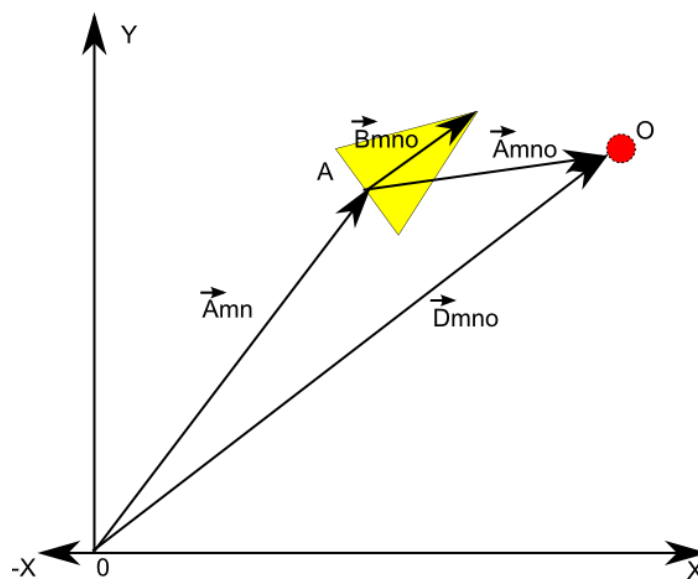
yang memiliki kode DET.

- Konversi besaran dan satuan. Besaran yang digunakan pada *buffer* masih berupa sudut estimasi dan durasi deteksi sonar. Sudut estimasi diproses dengan persamaan kalibrasi sudut sedangkan durasi deteksi sonar dikonversi kedalam satuan milimeter menggunakan persamaan-persamaan sebagaimana hasil kalibrasi dan penentuan besaran.

Setelah melalui proses penyaringan dan konversi besaran satuan, data dari *buffer* disimpan pada *data warehouse*. Tujuan proses konversi dan penyaringan ini adalah untuk menghasilkan tipe data yang seragam sesuai struktur tabel pada *data warehouse*. Sebagaimana pada rekapitulasi gerakan, rekapitulasi deteksi juga memiliki informasi tambahan yang merupakan penanda waktu ketika data diambil dari *buffer*.

Vektor deteksi obyek dinotasikan sebagai vektor \vec{A}_{mno} dan perlu diproses untuk menentukan posisi obyek terhadap titik acuan pada peta lokal. Pemrosesan dilakukan dua kali yaitu pengolahan posisi deteksi dikarenakan adanya simpangan sudut akibat arah agen yang senantiasa berubah baru dikalkulasi dengan posisi agen terhadap titik acuan pada peta lokal. Pemrosesan dilakukan agar seluruh vektor deteksi dapat diproyeksikan dari posisi awal yang sama sehingga deteksi dari posisi agen berbeda dapat diidentifikasi bila obyeknya sama.

Rekapitulasi posisi obyek relatif terhadap titik acuan pada peta lokal disimpan pada tabel *det_recap*. Posisi obyek pada peta lokal dinotasikan dalam vektor \vec{D}_{mno} yang menyimpan kedudukan polar agen terhadap titik acuan peta lokal. Untuk memahami konsep posisi obyek terhadap titik acuan pada peta lokal dapat diilustrasikan dengan gambar 3.12.



Gambar 3.12 Deteksi obyek

Gambar 3.12 menampilkan posisi obyek terhadap titik acuan merupakan simultan vektor deteksi obyek terhadap agen yang telah dikalkulasikan simpangannya terhadap arah agen dengan vektor posisi agen. Proses perhitungan simpangan dilakukan dengan memutar vektor deteksi obyek sejauh sudut pada vektor \vec{B}_{mn} yaitu sebesar β_{mn} . Proses selanjutnya dilakukan dengan menjumlahkan vektor deteksi dengan vektor posisi agen menggunakan konsep penjumlahan vektor sebagaimana dijelaskan pada dasar teori.

Untuk setiap \vec{A}_{mno} merupakan vektor deteksi obyek terhadap agen pada posisi \vec{A}_{mn} arah \vec{B}_{mn} dan α_{mno} merupakan komponen sudut dari vektor \vec{A}_{mno} dapat ditentukan \vec{B}_{mno} sebagai posisi obyek terhadap agen dengan posisi sudut yang telah disesuaikan dengan simpangan arah agen sehingga berlaku (3.5) dan (3.6).

$$\vec{D}_{mno} = \vec{A}_{mn} + \vec{B}_{mno} \quad (3.5)$$

$$\beta_{mno} = \alpha_{mno} + \beta_{mn} \quad (3.6)$$

Untuk l_d merupakan panjang dari vektor \vec{D}_{mno} dan α_d merupakan sudut dari vektor \vec{D}_{mno} sementara l_{mno} dan α_{mno} merupakan jarak dan arah deteksi obyek terhadap agen, sesuai kondisi pada narasi sebelumnya berlaku (3.7) dan (3.8).

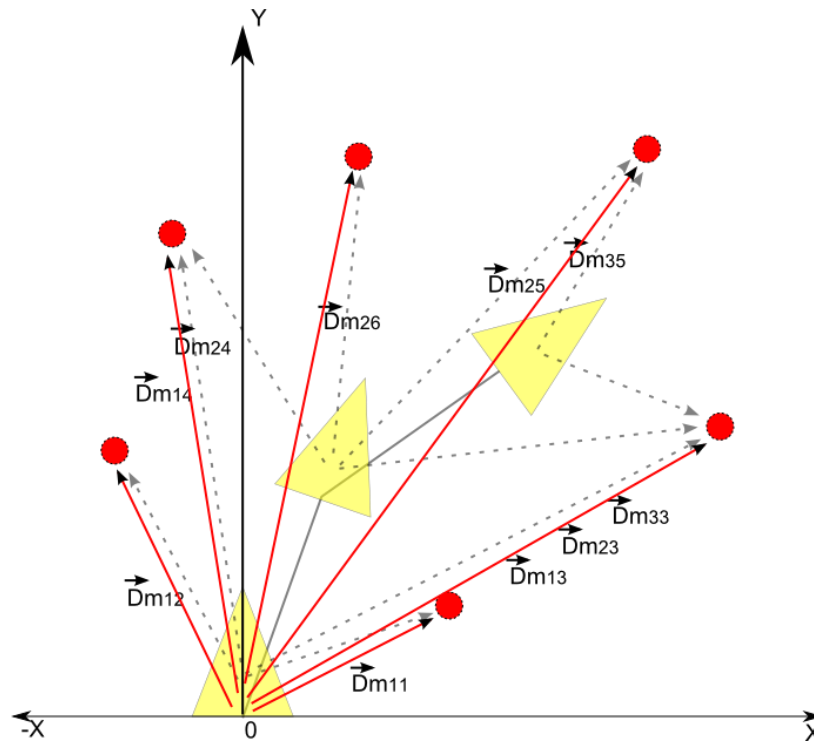
$$l_d = \sqrt{(\sum l_{dx})^2 + (\sum l_{dy})^2}$$

$$l_d = \sqrt{[l_{mn} \cos \alpha_{mn} + l_{mno} \cos(\beta_{mno} + \beta_{mn})]^2 + [l_{mn} \sin \alpha_{mn} + l_{mno} \sin(\beta_{mno} + \beta_{mn})]^2} \quad (3.7)$$

$$\alpha_d = \tan^{-1} \frac{(\sum l_{dy})}{(\sum l_{dx})}$$

$$\alpha_d = \tan^{-1} \frac{[l_{mn} \sin \alpha_{mn} + l_{mno} \sin(\beta_{mno} + \beta_{mn})]}{[l_{mn} \cos \alpha_{mn} + l_{mno} \cos(\beta_{mno} + \beta_{mn})]} \quad (3.8)$$

Proses perhitungan-perhitungan di atas menghasilkan satu besaran vektor sehingga memerlukan dua kolom variabel untuk penyimpanan yaitu satu besaran skalar jarak dan dua besaran skalar sudut. Ilustrasi dari rekap pergerakan robot ditampilkan pada gambar 3.13.



Gambar 3.13 Rekapitulasi Deteksi Obyek

Gambar 3.13 di atas menampilkan deteksi obyek yang ditampilkan sebagai bulatan merah pada setiap posisi agen yang ditampilkan sebagai segitiga kuning. Pada beberapa posisi berbeda terkadang agen mendeteksi obyek yang sama seperti pada posisi pertama, kedua dan ketiga sama-sama mendeteksi obyek pada posisi paling kanan yang pada gambar tampak tiga anak panah putus-putus ke arah obyek di paling kanan di rekap menjadi satu anak panah merah. Beberapa deteksi yang berulang menghasilkan vektor deteksi yang relatif sama pada rekapitulasi seperti $\overrightarrow{D_{m14}}$, $\overrightarrow{D_{m24}}$ juga $\overrightarrow{D_{m13}}$, $\overrightarrow{D_{m23}}$, $\overrightarrow{D_{m33}}$ yang pada kondisi ideal akan tepat sama satu sama lain.

3.6.6 Penyusunan Peta Lokal

Langkah awal dari penyusunan peta lokal adalah pengidentifikasian obyek. Di sisi deteksi obyek, proses ini merupakan bagian pertama dari layer OLAP. Pada proses ini dilakukan rekapitulasi posisi obyek berdasarkan sederetan hasil deteksi yang sudah direkapitulasi pada *data warehouse*.

Pada kondisi nyata, ketelitian sensor dan aktuator yang terbatas mengakibatkan deteksi obyek yang sama menghasilkan vektor yang berbeda sehingga keseluruhan posisi deteksi perlu diidentifikasi untuk mengestimasi posisi obyek. Proses identifikasi dilakukan dengan menganalisa posisi panjang dan sudut deteksi obyek terhadap titik acuan pada peta

lokal. Posisi deteksi dari step berbeda namun memiliki nilai yang relatif dekat dengan toleransi tertentu dianggap sebagai satu obyek yang sama. Besaran toleransi bisa diatur melalui konfigurasi yang tersedia pada antarmuka di sisi pengguna. Setelah ditemukan kesamaan, obyek kemudian diberi label identitas secara runtut berdasarkan waktu deteksi. Posisi obyek pada peta lokal dinotasikan sebagai $\overrightarrow{O_{mo}}$ dengan jarak l_{mo} pada sudut α_{mo} relatif terhadap titik acuan yang merupakan rata-rata dari seluruh posisi hasil deteksi. Tingkat keyakinan posisi sebuah obyek pada peta lokal dinyatakan sebagai W_{mo} yang merupakan jumlah deteksi obyek. Dengan asumsi satu kelompok $\overrightarrow{D_{mno}}$ merupakan deteksi dari obyek o yang sama berlaku (3.9) dan (3.10).

$$l_{mo} = \frac{\sum_{x=1}^{x=Nmno} l_{mno}}{W_{mo}} \quad (3.9)$$

$$\alpha_{mo} = \frac{\sum_{i=1}^{i=Nmno} \alpha_{mno}}{W_{mo}} \quad (3.10)$$

Sebagaimana dijelaskan pada tinjauan pustaka, peta pada penelitian ini dianalogikan sebagai peta pegas dengan informasi posisi obyek-obyek beserta relasi antara obyek satu dengan yang lain. Relasi obyek disusun sebagaimana rancangan solusi yang hendak diusulkan pada penelitian ini. Relasi antar obyek dapat diketahui dengan menghitung selisih besaran vektor antara posisi obyek satu dengan yang lain. Dengan asumsi koordinat polar posisi obyek pada peta lokal dianggap sebagai vektor, perhitungan relasi antar obyek dapat dilakukan dengan mengoperasikan 2 vektor posisi obyek dengan operasi pengurangan vektor.

Untuk setiap $\overrightarrow{R_{mp \rightarrow q}}$ merupakan posisi polar obyek q terhadap obyek p pada peta m dengan jarak dan sudut berturut-turut $l_{mp \rightarrow q}$ dan $\alpha_{mp \rightarrow q}$ dengan posisi obyek p dan q terhadap peta m berturut-turut $\overrightarrow{O_{mp}}$ dan $\overrightarrow{O_{mq}}$ berlaku (3.11) dan (3.12).

$$\begin{aligned} l_{mp \rightarrow q} &= \sqrt{(l_{mp \rightarrow qx})^2 + (l_{mp \rightarrow qy})^2} \\ l_{mp \rightarrow q} &= \sqrt{(l_{mqx} - l_{mpx})^2 + (l_{mqy} - l_{mpy})^2} \\ l_{mp \rightarrow q} &= \sqrt{(l_{mq} \cos \alpha_{mq} - l_{mp} \cos \alpha_{mp})^2 + (l_{mq} \sin \alpha_{mq} - l_{mp} \sin \alpha_{mp})^2} \end{aligned} \quad (3.11)$$

$$\begin{aligned} \alpha_{mp \rightarrow q} &= \tan^{-1} \frac{(l_{mp \rightarrow py})}{(l_{mp \rightarrow px})} \\ \alpha_{mp \rightarrow q} &= \tan^{-1} \frac{(l_{mqy} - l_{mpy})}{(l_{mqx} - l_{mpx})} \\ \alpha_{mp \rightarrow q} &= \tan^{-1} \frac{(l_{mq} \sin \alpha_{mq} - l_{mp} \sin \alpha_{mp})}{(l_{mq} \cos \alpha_{mq} - l_{mp} \cos \alpha_{mp})} \end{aligned} \quad (3.12)$$

Relasi antara agen dengan obyek juga dapat dikalkulasi dengan (3.11) dan (3.12) di

atas hanya saja yang dioperasikan antara koordinat polar posisi agen dan koordinat polar posisi obyek.

Tabel 3.1 Struktur informasi peta lokal

$l_{mn}, \alpha_{mn}, \beta_{mn}$	$l_{mn \rightarrow m1}, \alpha_{mn \rightarrow m1}$	$l_{mn \rightarrow m2}, \alpha_{mn \rightarrow m2}$		$l_{mn \rightarrow mo}, \alpha_{mn \rightarrow mo}$
$l_{m1 \rightarrow mn}, \alpha_{m1 \rightarrow mn}$	l_{m1}, α_{m1}	$l_{m1 \rightarrow m2}, \alpha_{m1 \rightarrow m2}$		$l_{m1 \rightarrow mo}, \alpha_{m1 \rightarrow mo}$
$l_{m2 \rightarrow mn}, \alpha_{m2 \rightarrow mn}$	$l_{m2 \rightarrow m1}, \alpha_{m2 \rightarrow m1}$	l_{m2}, α_{m2}		$l_{m2 \rightarrow mo}, \alpha_{m2 \rightarrow mo}$
			...	
$l_{mn \rightarrow mo}, \alpha_{mn \rightarrow mo}$	$l_{m1 \rightarrow mo}, \alpha_{m1 \rightarrow mo}$	$l_{m2 \rightarrow mo}, \alpha_{m2 \rightarrow mo}$		l_{mo}, α_{mo}

Posisi terkini agen, posisi obyek beserta relasi-relasinya disusun untuk proses mendukung keputusan perbandingan antara peta satu dengan yang lain dalam rangka menyusun peta global. Data relasi antar agen dan obyek dapat divisualisasikan dalam bentuk peta pegas sebagaimana dijelaskan pada tinjauan pustaka. Bentuk informasi rekapitulasi posisi agen dan obyek beserta relasi-relasinya ditampilkan pada tabel 3.1.

Relasi antara obyek q terhadap obyek p sebenarnya hanya merupakan kebalikan dari relasi antara obyek p terhadap obyek q. Vektor relasi obyek q terhadap obyek p dan relasi obyek p terhadap obyek q memiliki panjang yang sama namun dengan sudut berbeda 180^0 sehingga bisa berlaku (3.13), (3.14) dan (3.15) berikut.

$$R_{mp \rightarrow q} = -R_{mq \rightarrow p} \quad (3.13)$$

$$l_{mp \rightarrow q} = l_{mq \rightarrow p} \quad (3.14)$$

$$\alpha_{mp \rightarrow q} = \alpha_{mq \rightarrow p} + 180^0 \quad (3.15)$$

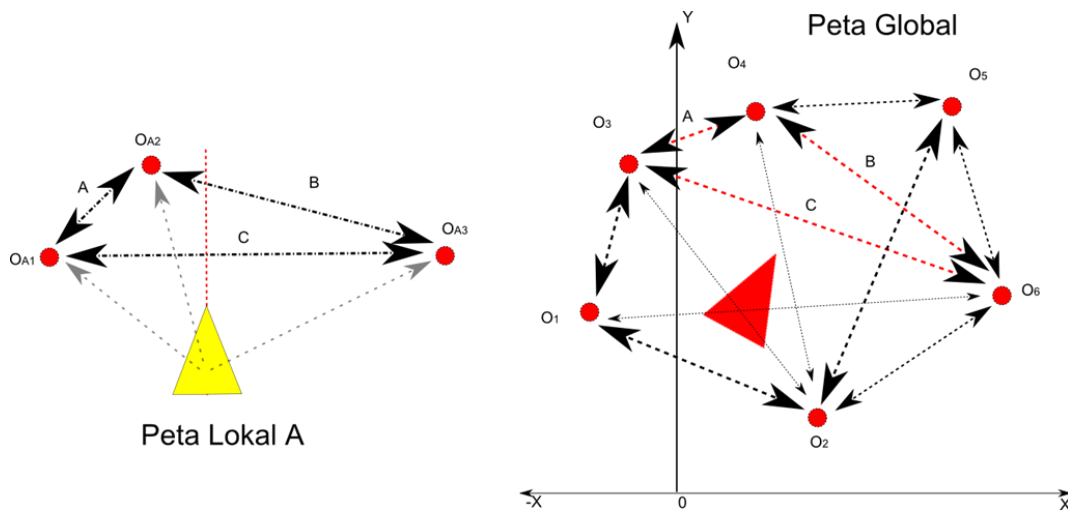
Persamaan (3.13), (3.14) dan (3.15) di atas menyatakan vektor posisi obyek yang satu dengan yang lain yang panjangnya tepat sama hanya arahnya saling berlawanan. Untuk efisiensi media penyimpanan, relasi posisi antar obyek hanya disimpan sekali yaitu relasi obyek dengan indeks yang lebih tinggi terhadap obyek dengan indeks yang lebih rendah.

3.6.7 Konsolidasi Peta Lokal dengan Peta Global

Setelah beberapa kali dilakukan percobaan dengan agen robot sederhana akan didapatkan rekapitulasi posisi agen dan robot pada beberapa peta lokal. Beberapa peta lokal ini dapat dikonsolidasi untuk menghimpun sebuah peta global. Sebelum dilakukan konsolidasi peta lokal, diperlukan sebuah besaran yang menyatakan relasi antara titik acuan

pada peta lokal yang satu dengan yang lain. Relasi antar peta ini digunakan untuk memetakan rekapitulasi posisi-posisi obyek ke satu bidang yang sama sehingga keseluruhan obyek yang telah terdeteksi dapat dikonsolidasi.

Pembandingan antara dua peta dilakukan dengan membandingkan posisi obyek-obyek yang ada pada peta lokal dengan posisi obyek-obyek pada peta global. Bentuk obyek yang seragam menjadikan sistem sulit untuk membedakan obyek yang satu dengan yang lain sehingga diperlukan pola identifikasi berdasarkan relasi antar obyek satu dengan sekitarnya. Serangkaian jarak antar beberapa obyek yang berdekatan dapat dijadikan acuan dalam pencarian obyek tersebut pada peta global. Untuk mempermudah pemahaman konsep diilustrasikan pada gambar 3.14.

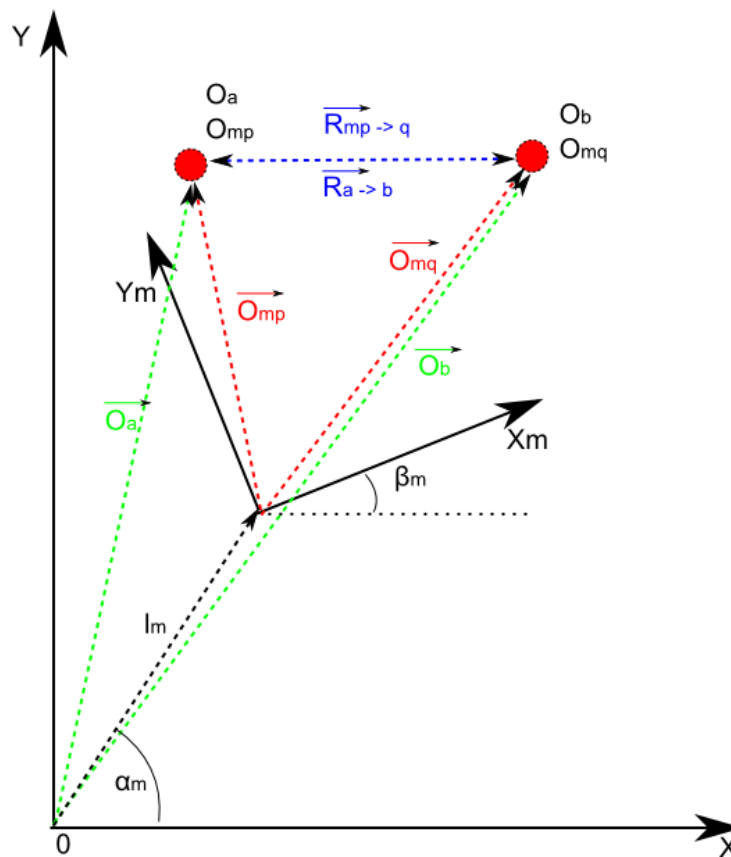


Gambar 3.14 Relasi antara peta lokal dan peta Global

Gambar 3.14 di atas menampilkan posisi agen pada peta lokal sedang mendeteksi obyek O_{A1} , O_{A2} dan O_{A3} yaitu obyek 1,2 dan 3 pada peta lokal A. Setelah dilakukan proses rekapitulasi obyek dan penyusunan peta lokal diketahui bahwa jarak antara O_{A1} dan O_{A2} sejauh A, jarak antara O_{A2} dan O_{A3} sejauh B, dan jarak antara O_{A1} dan O_{A3} sejauh C. Tiga besaran jarak yang didapatkan pada peta lokal digunakan sebagai filter pencarian pada peta global sedemikian rupa sehingga ditemukan tiga obyek yang memiliki relasi jarak serupa. Bila pada peta global ditemukan lebih dari satu kelompok obyek atau pencarian belum membuahkan hasil, maka proses penelusuran pada peta lokal dilanjutkan hingga ditemui obyek keempat sebagai acuan tambahan.

Ketika ditemui relasi antar obyek pada peta lokal dan peta global, obyek-obyek pada peta lokal tersebut dipetakan pada peta global. Obyek yang pertama kali dipetakan adalah beberapa obyek yang menjadi filter pencarian yaitu minimal tiga obyek. Identitas ketiga

obyek lokal yang sudah dipetakan pada obyek global disimpan dalam sebuah tabel relasi. Dua peta lokal berbeda dapat memuat obyek yang sama tergantung obyek yang dideteksi sepanjang rute yang dilalui agen. Contoh konsolidasi peta lokal L pada peta global dapat diilustrasikan pada gambar 3.15.



Gambar 3.15 Konsolidasi peta lokal M pada peta global

Gambar 3.15 di atas menampilkan obyek p dan q pada peta lokal M yang kebetulan merupakan obyek a dan b pada peta global. Posisi obyek terhadap peta lokal digambarkan sebagai garis putus-putus merah yang pada gambar diberi notasi \vec{O}_{mp} dan \vec{O}_{mq} sementara posisi obyek terhadap peta global digambarkan sebagai garis putus-putus hijau dengan notasi \vec{O}_a dan \vec{O}_b . Posisi relatif peta lokal terhadap peta global ditampilkan dalam garis putus-putus hitam. Untuk mengetahui posisi relatif peta lokal terhadap peta global diperlukan minimal dua posisi obyek pada peta lokal dan global. Relasi antara kedua obyek tersebut pada peta lokal dinyatakan sebagai $\vec{R}_{mp \rightarrow q}$ dan pada peta global dinyatakan sebagai $\vec{R}_{a \rightarrow b}$ yang pada gambar ditampilkan sebagai garis putus-putus biru. Kedua vektor relasi ini memiliki panjang yang sama karena memang posisi obyeknya sama, hanya saja sudut vektornya berbeda dikarenakan adanya perbedaan orientasi peta lokal M dan peta global. Untuk semua obyek p dan q pada peta lokal m yang merupakan obyek a dan b pada peta global dan

merupakan relasi antara peta lokal m terhadap peta global berlaku (3.16), (3.17) dan (3.18) berikut.

$$\beta_m = \alpha_{mp \rightarrow q} - \alpha_{a \rightarrow b} \quad (3.16)$$

$$l_m = \sqrt{(l_{mx})^2 + (l_{my})^2}$$

$$l_m = \sqrt{[l_a \cos \alpha_a - l_{mp} \cos(\alpha_{mp} - \beta_m)]^2 + [l_a \sin \alpha_a - l_{mp} \sin(\alpha_{mp} - \beta_m)]^2} \quad (3.17)$$

$$\alpha_m = \tan^{-1} \left(\frac{l_{my}}{l_{mx}} \right)$$

$$\alpha_m = \tan^{-1} \left(\frac{l_a \sin \alpha_a - l_{mp} \sin(\alpha_{mp} - \beta_m)}{l_a \cos \alpha_a - l_{mp} \cos(\alpha_{mp} - \beta_m)} \right) \quad (3.18)$$

Persamaan (3.16), (3.17) dan (3.18) di atas menghasilkan tiga besaran yaitu satu besaran skalar jarak titik acuan peta lokal M terhadap titik acuan peta global, satu besaran sudut posisi titik acuan peta lokal M terhadap titik acuan peta global, serta satu besaran sudut orientasi titik acuan peta lokal M terhadap titik acuan peta global. Ketiga besaran yang dihasilkan kemudian disimpan sebagai relasi antara peta lokal dan peta global. Besaran-besaran ini yang kemudian akan dijadikan acuan pemetaan seluruh obyek pada peta lokal ke peta global sehingga seluruh obyek dapat terkonsolidasi pada peta global.

Relasi antara peta lokal dengan peta global dapat digunakan untuk memetakan seluruh obyek yang ada pada peta lokal ke dalam peta global sehingga seluruh posisi obyek pada area observasi dapat dipetakan. Seluruh obyek kemudian kembali diteliti posisinya dan dengan rentang toleransi tertentu dikelompokkan untuk mengidentifikasi obyek yang sama. Obyek-obyek pada peta lokal dapat dipetakan pada peta global dengan menggunakan persamaan simultan vektor sebagaimana digunakan pada rekapitulasi deteksi obyek.

Setelah byek-obyek pada peta lokal dipetakan pada peta global, obyek-obyek tersebut kembali dianalisa relasi-relasinya. Relasi antar obyek pada peta global digunakan sebagai pembandingan untuk menentukan relasi antara peta global dengan peta lokal pada observasi-observasi berikutnya.

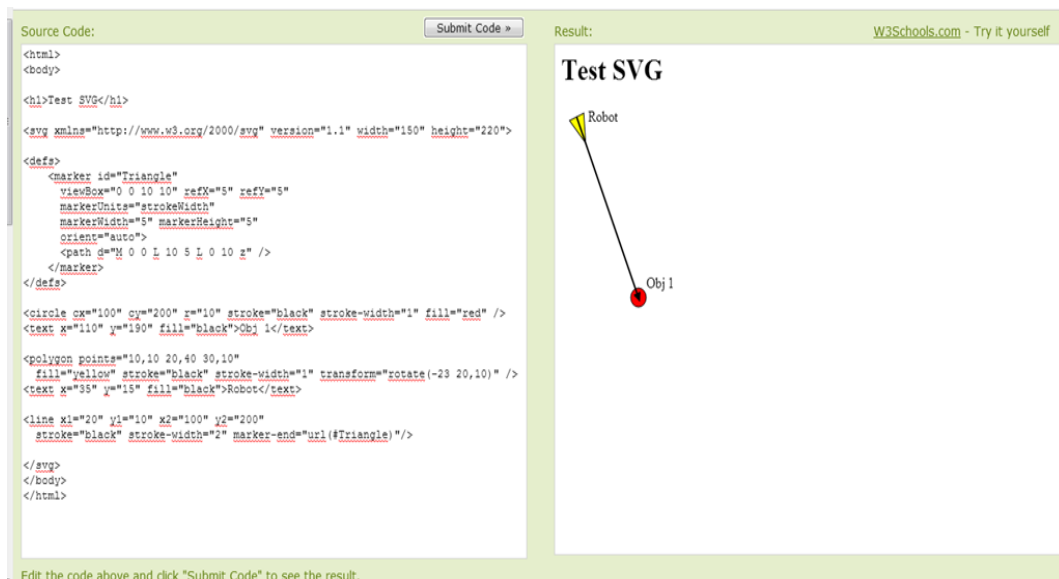
3.7 Rancang Bangun Antarmuka

Tahapan penelitian selanjutnya adalah rancang bangun antarmuka. Perancangan antarmuka sistem dan pengguna direncanakan sesederhana mungkin dengan berbasis web agar memudahkan akses pengguna dari berbagai media. Informasi yang perlu ada untuk ditampilkan di sisi pengguna antara lain sebagai berikut :

- Menu akses ke berbagai komponen pada antarmuka.
- Tampilan peta global berisi posisi obyek dan robot.
- Tampilan peta pegas berisi relasi antar obyek dan robot.
- Tampilan rekam pergerakan robot pada peta global.
- Tampilan peta lokal berisi posisi obyek-obyek hasil deteksi relatif terhadap robot.

Antarmuka sistem ke pengguna direncanakan untuk dibangun dengan bahasa pemrograman PHP dengan hasil keluaran berupa tag-tag HTML. Tampilan HTML ini kemudian diakses oleh pengguna melalui peramban secara interaktif dengan berbagai menu dan tampilan yang ada.

Visualisasi peta pada antarmuka dirancang dengan menggunakan standar SVG. Pemilihan standar SVG dikarenakan standar tersebut berbasis XML sehingga mudah dikomposisi menggunakan bahasa PHP dan dapat diakses oleh berbagai macam peramban. Contoh visualisasi dengan SVG ditampilkan pada gambar 3.16.



Gambar 3.16 Visualisasi dengan SVG

Gambar 3.16 di atas menampilkan contoh hasil visualisasi dengan menggunakan SVG. Ilustrasi robot berbentuk segitiga dikomposisi dengan fungsi *polygone* sementara ilustrasi obyek berbentuk lingkaran dikomposisi dengan fungsi *circle*. Deteksi obyek diilustrasikan dengan anak panah yang dikomposisi dengan fungsi *line*.

3.8 Pengujian Sistem

Tahapan selanjutnya adalah pengujian sistem. Pengujian sistem pada penelitian ini meliputi dua tahapan yaitu *unit testing*, dan *system testing*. Pada *unit testing* setiap komponen di uji serta di kalibrasi ulang untuk meminimalisir selisih akibat *noise*. Beberapa tahapan yang dilakukan pada *unit testing* diantaranya sebagai berikut.

- Uji deteksi sensor ultrasonik
- Uji komunikasi serial melalui transmisi *bluetooth*
- Uji penyimpanan data telemetri ke dalam basis data
- Uji ekstraksi data ke *data warehouse*
- Uji query informasi dari *data warehouse* ke OLAP
- Uji visualisasi data OLAP

Pada *system testing* keseluruhan modul diuji secara bersamaan baik menggunakan simulasi maupun data nyata. Hasil pengujian dibandingkan dengan kondisi informasi pada basis data untuk memastikan kebenaran informasi yang ditampilkan sistem apakah sudah sesuai dengan data yang tersimpan pada basis data. Pengujian sistem dilakukan dua kali diantaranya sebagai berikut.

- Pengujian simulasi. Pengujian simulasi dilakukan dengan menggunakan data simulasi, bukan hasil deteksi sebenarnya dari robot. Proses pengujian dilakukan dengan memasukkan data-data hasil pengukuran dan perhitungan ideal ke dalam sistem. Pengujian dengan simulasi ini difungsikan sebagai pembuktian konsep bahwa sistem dapat berfungsi dengan baik sesuai asumsi yang diambil.

- Pengujian robot. Pengujian robot dilakukan dengan menggunakan hasil deteksi robot sesuai kondisi nyata. Proses pengujian dilakukan dengan mengintegrasikan robot dengan sistem menggunakan antarmuka yang telah dibuat. Pengujian ini kemudian dibandingkan dengan hasil pengujian simulasi.

DAFTAR PUSTAKA

1. Badros, G.J., Tirtowidjojo, J.J., Marriot, K., Meyer, B., Portnoy, W., dan Borning, A., 2001, A Constraint Extension to Scalable Vector Graphics, *Proceeding of the 10th International Conference on World Wide Web*, May 1-5, 489-498.
2. Bailey, T., dan Durrant-Whyte, H., 2006, Simultaneous Localisation and Mapping (SLAM) Part II : State of the Art, *IEEE Robotics & Automation Magazine Vol 13 issue 3*, 108 - 117.
3. Bao, Y., dan Zhang, L.J., 2010, Decision Support System Based on Data Warehouse, *World Academy of Science, Engineering and Technology 47*, 172-176.
4. Bimonte, S., dan Tchounikine, A., 2007, Spatial OLAP: Open Issues and a Web Based Prototype, *10th AGILE International Conference on Geographic Information Science*, Aalborg.
5. Cao, Y., dan Ames, D.P., 2010, Towards Virtual Watersheds: Integrated Data Mining, Management, Mapping and Modeling, *AWRA 2010 Spring Specialty Conference*, Orlando, FL, March 29-31.
6. Chan, J.O., 2005, Optimizing Data Warehousing Strategies, *Communications of the IIMA Vol 5 issue 1*, 1-14.
7. Chaudhuri, S., dan Dayal, U., 1997, An Overview of Data Warehousing and OLAP Technology, *ACM Sigmod Record*, 517-526.
8. Chong, K.S., dan Kleeman, L., 1999, Feature-based Mapping in Real Large Scale Environment using an Ultrasonic Array, *The International Journal of Robotic Research Vol 18 no 1*, 3-19.
9. Dahlstrom, E., Dengler, P., Grasso, A., Lilley, C., McCormack, C., Schepers, D., Watt, J., Ferraiolo, J., Jun, F., dan Jackson, D., 2011, *Scalable Vector Graphics 1.1 Second Edition*, W3C Recommendation.
10. Dennis, A., Wixom, B.H., dan Tegarden, D., 2005, *Systems Analysis and Design with UML Version 2.0 : An Object-Oriented Approach*, John Wiley & Sons, Hoboken, NJ.
11. Druzdzel, M.J., dan Flynn, R.R., 2002, *Decision Support System. In: Kent, Allen(ed), Encyclopedia of Library and Information Science, Second Edition*, Marcel Dekker, New York, NY.
12. Duckett, T., 2003, A Genetic Algorithm for Simultaneous Localization and Mapping, *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, Taipei, September 14-19, 434-439.
13. Durrant-Whyte, H., dan Bailey, T., 2006, Simultaneous Localisation and Mapping (SLAM) Part I : The Essential Algorithms, *IEEE Robotics & Automation Magazine Volume 13 issue 2*, 99-110.
14. Favinger, A., 2008, *The Polar Coordinate System*. Master thesis, University of Nebraska-Lincoln.
15. Hammergren, T.C., dan Simon, A.R., 2009, *Data Warehousing for Dummies 2nd Edition*, John Wiley & Sons, Hoboken, NJ.
16. Ma, Y., Kalashnikov, dan D.V., Mehrotra, S., 2008, Toward Managing

Uncertain Spatial Information for Situational Awareness Applications, *IEEE Transactions on Knowledge and Data Engineering* vol 20 no 10, 1-16.

17. Pal, S., dan Tripathy, N.S., 2011, Remote Position Control System of Stepper Motor using DTMF Technology, *International Journal of Control and Automation* vol 4 no 2, 35-42.

18. Peacock, J., 2009, *Foundation of Mathematical Physics : Vector, Tensor and Fields*. Teaching Material, University of Edinburg.

19. Pradhana, H.W., Suryono dan Widodo, A., 2013, Visualization of Simultaneous Localization and Mapping using SVG, *Proceeding of The 2nd International Conference on Information System for Business Competitiveness*, Semarang, Desember 5, 134-138.

20. Pradhana, H.W., Widodo, A., dan Suryono., 2013, Web Based Map Generations of Mobile Robot Movement using Scalable Vector Graphic, *Proceeding of The 1st Conference on Information Technology, Computer, and Electrical Engineering*, Semarang, November 16, 23-27.

21. Pratama, P.S., dan Kim, S.B., 2013, *Obstacle Avoidance Algorithm Based on Velocity and Orientation Controls of Differential Drive Automatic Guided Vehicle*. Master thesis, Pukyong National University.

22. Ruckert, E.A., Bischof, H., dan Ruther, M., 2009, *Simultaneous Localisation and Mapping for Mobile Robots with Recent Sensor Technologies*. Master thesis, Graz University of Technology.

23. Saleem, M.M., 2013, An Economic Simultaneous Localization and Mapping System for Remote Mobile Robot Using SONAR and an Innovative AI Algorithm, *International Journal of Futur Computer and Communication* vol 2 no 2, 147-150.

24. Sen, A., dan Sinha, A.P., 2005, A Comparison of Data Warehousing Methodologies, *Communications of the ACM* Vol 48, 79-84.

25. Singh, M., Singh, R., dan Singh, B., 2010, Microcontroller Based Clockwise/Anticlockwise Stepper Motor Controller using PC Keyboard via Com Port, *International Journal of Computer Science & Communication* vol 1 no 1, 189-191.

26. Strang, G., 1991, *Calculus*, Wellesley-Cambridge Press, Wellesley, MA.

27. Turkmen, G., Yazici, A., dan Cagiltay, N.E., 2007, *Developing a Data Warehouse for University Decision Support System*, Master Thesis, Computer Engineering of Atilim University.