

**SISTEM *MAPPING* JALUR ROBOT BERDASARKAN DATA
MIKROKONTROLER UNTUK MEMETAKAN LINGKUP
GERAK ROBOT MENGGUNAKAN
METODE EKF-SLAM**

SKRIPSI

Oleh:

GERY ARIES SUKMA WIBOWO

NIM. 10650096



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**SISTEM *MAPPING* JALUR ROBOT BERDASARKAN DATA
MIKROKONTROLER UNTUK MEMETAKAN LINGKUP
GERAK ROBOT MENGGUNAKAN
METODE EKF-SLAM**

SKRIPSI

Oleh:

GERY ARIES SUKMA WIBOWO

NIM. 10650096



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

HALAMAN PENGAJUAN

**SISTEM *MAPPING* JALUR ROBOT BERDASARKAN DATA
MIKROKONTROLER UNTUK MEMETAKAN LINGKUP
GERAK ROBOT MENGGUNAKAN
METODE EKF-SLAM**

SKRIPSI

**Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
GERY ARIES SUKMA WIBOWO
NIM. 10650096 / S - 1**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2014**

HALAMAN PERSETUJUAN

**SISTEM *MAPPING* JALUR ROBOT BERDASARKAN DATA
MIKROKONTROLER UNTUK MEMETAKAN LINGKUP
GERAK ROBOT MENGGUNAKAN
METODE EKF-SLAM**

SKRIPSI

Oleh :

Nama : Gery Aries Sukma Wibowo
NIM : 10650096
Jurusan : Teknik Informatika
Fakultas : Sains Dan Teknologi

Telah Diperiksa dan Disetujui untuk Diuji:

Tanggal :

Dosen Pembimbing I

Dosen Pembimbing II

Yunifa Miftachul Arif, M. T.
NIP. 19830616 201101 1 004

Irwan Budi Santoso, M. T.
NIP. 19770103 201101 1 004

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

SURAT PERNYATAAN
ORISINALITAS PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : Gery Aries Sukma Wibowo
NIM : 10650096
Fakultas/Jurusan : Sains Dan Teknologi / Teknik Informatika
Judul Penelitian : *System Mapping Jalur Robot Berdasarkan Data Mikrokontroller Untuk Memetakan Lingkup Gerak Robot Menggunakan Metode EFF-SLAM*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut

Malang, November 2014
Yang Membuat Pernyataan,

Gery Aries Sukma Wibowo
10650096

Motto



HALAMAN PERSEMBAHAN

Alhamdulillah...

Segala Puji bagi Allah SWT Tuhan semesta alam, kupersembahkan sebuah karya sederhana untuk orang-orang yang paling aku sayangi dan aku banggakan,

Ayah dan Ibu

SETIYO WDODO dan SRI HANDAYANI

Atas Keikhlasannya dalam mendo' akanku,
Atas Kerendahan hatinya dalam membimbingku,
Atas ketulusannya mengingatkan ku dalam kebaikan,
Atas Segalanya yang tak mungkin semua ku ungkapkan,
hanya dalam tulisan sederhana

Adik – adiku

Gilang Aries Setyo Pamungkas dan Gading Aries Al-Kindi

Yang selalu membuatku semangat dan tak pernah lupa untuk mendoakanku

Seluruh keluarga besarku

Nenek, kakak ipar, dan keponakan

Semoga Allah SWT melindungi dan menjaga kalian, Aamiin...

Dan ucapan terimakasih yang sebanyak-banyaknya kepada :

Sahabat seperjuangan, akhirnya kita bisa menyelesaikan karya ini

Sahabat – sahabatku yang kubanggakan An' im Fahmi, Novi Anto, Moch. Arief, A. Tyas Muzakki, Ali, Acip Syaifudin, dan Taufan Nur Hidayat yang selalu memberi semangat, membantu, mendukung, dan mendo' akan.

Teman – teman MMM (Vivid, Novi, Naufal, Zaenal, Dewi, Afif, Dzikri, Syafei, Agus, Dian, Dita, Ade, dan Haris) yang selalu memberi semangat, dukungan, serta menjadi tempat berbagi suka dan duka

Teman - teman yang mau membagi ilmunya dan tak lelah untuk mengajari yaitu Taufan dan Juniardi

Teman - teman seperjuangan Sari, Vina, Kiki, Balqis, Wati, Riris, Amel, Fuad, Amru dan An' im, sukses untuk kita semua

Teman - teman 4GTI , belajar bersama dari awal perkuliahan hingga saat ini

Teman - teman Infinity (TI 2010) , kalian sudah menjadi keluarga bagiku



KATA PENGANTAR



Segala puji bagi Allah SWT yang telah melimpahkan rahmat, hidayah serta inayah-Nya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “*Sistem Generate Map Berdasarkan Sensor Jarak Untuk Memetakan Daerah yang Sulit Dijangkau Menggunakan Metode EKF-SLAM*”.

Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW yang telah membimbing umatnya dari gelapnya kekufuran menuju cahaya Islam yang terang benderang.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, oleh karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati patutlah penulis ucapkan terima kasih kepada:

1. Prof. DR. H. Mudjia Rahardjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Bayyinatul Muchtaromah, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crys dian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Yunifa Miftachul Arif, M. T. dan Irwan Budi Santoso, M. T selaku dosen pembimbing I dan II yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.

5. Segenap sivitas akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
6. Bapak dan Ibuku tercinta, adikku-adikku dan seluruh keluarga besar yang senantiasa memberikan doa dan restunya kepada penulis dalam menuntut ilmu serta dalam menyelesaikan skripsi ini.
7. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan, penulis ucapkan terima kasih yang sebesar-besarnya.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna, untuk itu penulis selalu menerima segala kritik dan saran dari pembaca. Harapan penulis, semoga karya ini bermanfaat bagi kita semua.

Malang, November 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGAJUAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN.....	v
MOTTO.....	vi
HALAMAN PERSEMBAHAN.....	vii
HALAMAN PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
ABSTRAK.....	xiv
ABSTRACT.....	xv
الخلاصة.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	6
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian.....	7
1.5 Manfaat Penelitian.....	7
1.6 Sistematika Penulisan.....	7
BAB II KAJIAN PUSTAKA.....	9
2.1 Peta Dalam Pandangan Islam.....	9
2.2 Simultaneous localization and Mapping (SLAM).....	11
2.3 Kompleksitas SLAM.....	12

2.4 Metode Dalam SLAM.....	13
2.5 Problem Teknis SLAM.....	13
2.5.1 Mapping.....	14
2.5.2 Sensing.....	15
2.5.3 Locating.....	16
2.5.4 Modeling.....	17
2.5.5 Exploration.....	17
2.6 Extended Kalman Filter (EKF-SLAM).....	18
2.7 Algoritma EKF-SLAM.....	19
2.8 Aplikasi Hybird.....	19
2.9 Phonegap Cordova	21
2.10 Processing.....	23
2.11 Android.....	24
2.12 Bahasa Pemrograman Java.....	26
2.13 Arduino.....	26
2.14 Bluetooth.....	28
BAB III PERANCANGAN SISTEM.....	31
3.1 Analisa dan Perancang Sistem.....	31
3.1.1 Prinsip Kerja Sistem.....	33
3.1.2 Perancangan Sistem.....	34
3.1.2.a Mockup.....	34
3.1.2.b Input.....	36
3.1.2.c Media Transmisi.....	34
3.1.2.d Output.....	37
3.1.3 Flowchart Aplikasi Mobile.....	39
3.2 Proses EKF-SLAM.....	40

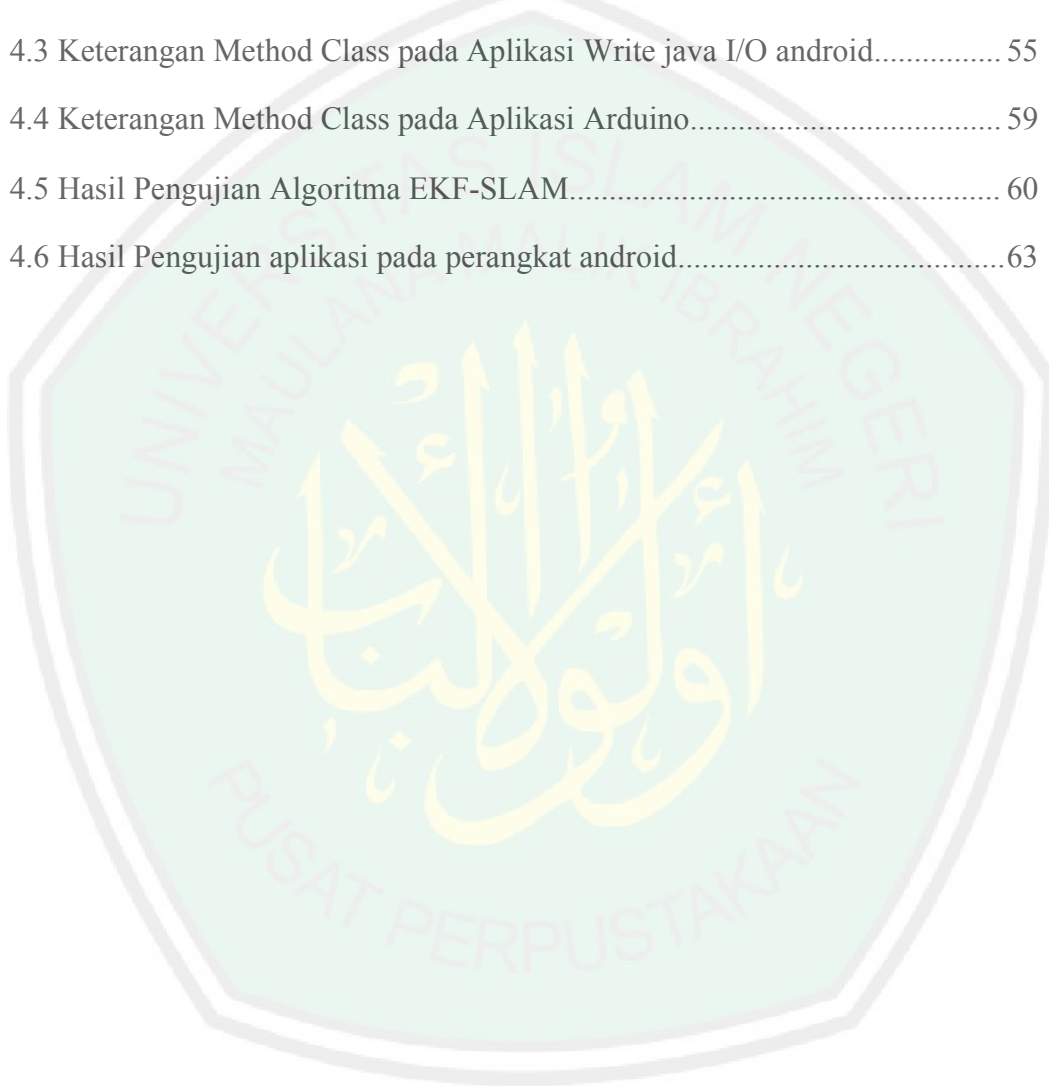
3.3 Algoritma <i>Extended Kalman Filter (EKF-SLAM)</i>	41
3.4 Kebutuhan Sistem.....	44
3.4.1 Kebutuhan Perangkat Keras (<i>Hardware</i>).....	44
3.4.2 Kebutuhan Perangkat Lunak (<i>Software</i>).....	44
3.4.3 Kebutuhan <i>Device</i> Minimum.....	46
BAB IV HASIL DAN PEMBAHASAN.....	47
4.1 Implementasi.....	47
4.2 Implementasi Algoritma <i>EKF-SLAM</i> pada Aplikasi.....	47
4.2.1 Program Processing ekfslam.pde.....	48
4.2.2 Program Map.html.....	50
4.2.3 Android Phonegap Injeksi data Java I/O.....	51
4.2.4 Sistem Mikrokontroler Simulator <i>Vechile</i> untuk Mengirim Data.....	57
4.2.5 Program untuk Mengirim Data di Arduino Uno R3.....	58
4.3 Hasil pengujian Algoritma <i>EKF-SLAM</i> pada Aplikasi.....	59
4.4 Implementasi Aplikasi.....	61
4.5 Pengujian Aplikasi pada Perangkat <i>Android</i>	65
4.6 Integrasi Dalam Islam.....	66
BAB V PENUTUP.....	68
5.1 Kesimpulan.....	68
5.2 Saran.....	69
DAFTAR PUSTAKA.....	
LAMPIRAN.....	

DAFTAR GAMBAR

2.1 Arsitektur Android.....	25
2.2 Mikrokontroler Board.....	27
2.3 Bluetooth Module.....	30
3.1 Rancangan Sistem Aplikasi.....	31
3.2 Diagram Blok Sistem.....	33
3.3 Mockup Tampilan Depan.....	34
3.4 Mockup Tampilan Perizinan Bluetooth.....	35
3.5 Mockup Simulasi Peta.....	35
3.6 Rangkaian Mikrokontroller	37
3.7 Output Aplikasi 1.....	38
3.8 Output Aplikasi 2.....	38
3.9 Flowchart Aplikasi Bluetooth.....	39
3.10 Proses EKF-SLAM.....	40
3.11 Manual EKF-SLAM 1.....	42
3.12 Manual EKF-SLAM 2.....	42
3.13 Arduino UNO Board.....	44
3.14 DFR Bluetooth V3.....	45
3.15 Evercoss A26C.....	45
4.1 Diagram Blok Aplikasi Writer I/O java.....	47
4.2 Sistem Mikrokontroller dan Module Bluetooth.....	57
4.3 Tampilan Depan Aplikasi.....	61
4.4 Tampilan Awal Aplikasi.....	62
4.5 Tampilan Saat Simulasi Berjalan.....	62
4.6 Tampilan Simulasi Selesai.....	63

DAFTAR TABEL

2.1 Ponegap Suported Platform.....	22
3.2 Kebutuhan Device.....	46
4.1 Keterangan Method Class pada Aplikasi processing.....	49
4.2 Keterangan Method Class pada aplikasi Android Bluetooth dan Start.....	52
4.3 Keterangan Method Class pada Aplikasi Write java I/O android.....	55
4.4 Keterangan Method Class pada Aplikasi Arduino.....	59
4.5 Hasil Pengujian Algoritma EKF-SLAM.....	60
4.6 Hasil Pengujian aplikasi pada perangkat android.....	63



ABSTRAK

Wibowo, Gery Aries Sukma. 2014. *Sistem Mapping Jalur Robot Berdasarkan Data Mikrokontroller untuk Memetakan Lingkungan Gerak Robot Menggunakan Metode EKF-SLAM*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) Yunifa Miftachul Arif, M. T. (II) Irwan Budi Santoso, M. T

Kata Kunci: *Mapping, Simulasi, EKF-SLAM, Mobile Mapping, Simultaneous Localization and Mapping*

Sistem yang dapat menggambarkan lingkungan gerak robot pada aplikasi *mobile* seperti *smartphone android* akan sangat memudahkan dalam mengamati suatu wilayah. Salah satu metode dalam memvisualisasikan peta dengan baik adalah *Extended Kalman Filter* (EKF-SLAM). Penanaman metode EKF-SLAM dibutuhkan untuk dapat membangun sebuah peta yang akurat. Penelitian ini membahas tentang membangun sebuah peta berdasarkan data yang dikirimkan oleh mikrokontroler sebagai simulator *vehicle* atau robot ke perangkat *smartphone android* untuk kemudian data tersebut diolah menjadi sebuah peta. Dari uji coba yang telah peneliti lakukan, dari simulasi algoritma EKF-SLAM peneliti memasukkan input parameter dari mikrokontroler *arduino* untuk dikirim ke dalam aplikasi *mapping* pada perangkat *android*. Aplikasi berjalan menggunakan platform *phonegap* yang berbasis *mobile web*. Data input parameter untuk EKF-SLAM di-*inject* menggunakan *java I/O* sebagai *writer data*. Data yang dikirim ke perangkat *android* harus disiapkan terlebih dahulu sebelum peta dijalankan dikarenakan proses menulis data pada *file pde thread* perlu di *close* terlebih dahulu. Saat proses *close thread* tersebut aplikasi *android* akan mengalami *Crash* pada *UI Thread Phonegap*. Dengan kondisi tersebut data harus diproses satu per satu. Dalam penelitian ini input koordinat dipersiapkan terlebih dahulu, setelah data siap maka data sudah dapat dibaca oleh aplikasi simulasi peta ini.

ABSTRACT

Wibowo, Gery Aries Sukma. 2014. *Path Robot Mapping System Based on Microcontroller Data for Create Robot Movement and Enviroment Using EKF-SLAM*. Thesis. Informatics Department of Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University, Malang.

Adviser: (I) Yunifa Miftachul Arif, M. T. (II) Irwan Budi Santoso, M. T

Keywords: *Mapping, Simulation, EKF-SLAM, Mobile Mapping, Simultaneous Localization and Mapping*

The system can generate maps on mobile applications such as smartphones android will greatly facilitate the observed region. One good method to visualize the map is Extended Kalman Filter (EKF-SLAM). Planting EKF-SLAM method is needed to be able to build an accurate map. This study discusses building a map based on data transmitted by the microcontroller as vechile or robot simulator for android smartphone device and then the data is processed into output map. From experiments which have researchers did, from EKF-SLAM algorithm simulation researchers are trying to enter the input parameters using arduino microcontroller to be sent to the generated map application. Applications run using PhoneGap platform-based mobile web. Data input parameters for the EKF-SLAM injected using java I / O as a writer of data. Data sent to the android device must be prepared in advance before the map is executed. Once the process of writing data to a file .pde, system needs to close the thread first. When the process close the thread will make Crash on android Phonegap application UI Thread. Under these conditions the data must be processed one by one. In this study, the input coordinates prepared first, after the data is ready, the data is readable by this map simulation applications.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indikator baru dalam perekonomian berkelanjutan yang menyebutkan bahwa salah satu faktor investasi adalah pengetahuan, yaitu pendidikan tinggi, kegiatan litbang, dan ICT (informasi, komunikasi dan telekomunikasi). Dengan demikian harus lebih diperhatikan pengembangan SDM yang berkualitas yang dihasilkan oleh perguruan tinggi, serta litbang yang terkait dengan bidang perekayasa robot, yang tidak lepas dari bidang elektronika, manufacturing, komputasi, otomasi kendali peralatan dan proses produksi industri.

Dengan semakin berkembangnya iptek, maka robot akan sangat bermanfaat apabila digunakan pada tempat / lingkungan yang berbahaya bagi manusia, atau yang membutuhkan tenaga yang sangat kuat, atau pada industri manufaktur yang kompleks dengan ketepatan dan presisi tinggi. Pada pengembangannya, robot telah digunakan pula pada penelitian-penelitian laut dalam, ruang angkasa, fenomena alam yang beresiko tinggi, dll. Perkembangan terkini adalah dalam aplikasi nanoteknologi, yang perlu diperhatikan terutama dalam hal otomasi kendali peralatan dan instrumentasi. (Semiloka Perkembangan Robot di Indonesia 2004: *"Peluang dan Tantangan Teknologi Robot di Indonesia"*)

Robot yang digunakan dalam penelitian-penelitian laut dalam, ruang angkasa, dan fenomena alam beresiko tinggi telah memanfaatkan banyak sensor

untuk mengenali lingkungan dan dirinya sendiri. Sensor-sensor tersebut merupakan alat yang berfungsi sebagai masukan data pada robot, sehingga data yang diperoleh dari sensor tersebut dapat diolah menjadi sebuah informasi. Sepertihalnya indera peraba manusia yang mengirim informasi ke otak tentang benda apa yang disentuhnya.

Berdasarkan informasi diatas mendasari penelitian ini untuk membangun aplikasi sistem generate map untuk daerah yang sulit dijangkau berdasarkan parameter data sensor yang dikirim oleh robot dengan menerapkan metode Extende Kalman Filter Simultaneous localization and Mapping (EKF-SLAM). Sehingga dengan adanya aplikasi ini diharapkan dapat membantu dalam mengamati daerah yang sulit dijangkau ataupun yang belum terjamah oleh manusia dengan bantuan robot. Dengan adanya aplikasi mobile monitoring control maka dapat dihasilkan system kendali yang lebih cepat dan *real time*, terutama dalam menyikapi dunia teknologi saat ini yang menuntut kualitas, kecepatan dan ketepatan.

Merupakan hal yang sudah diketahui oleh kebanyakan kaum muslimin, terlebih lagi oleh para penuntut ilmu agama, keutamaan besar yang Allah disediakan bagi orang-orang yang mempelajari ilmu. Namun sayangnya, kebanyakan dari kita, termasuk para penuntut ilmu sendiri sering lalai dan kurang menyadari bahwa ilmu yang dimaksud dalam ayat-ayat Al Qur-an dan hadits-hadits Nabi tersebut bukanlah sekedar teori belaka, yang hanya terlihat dalam bentuk hapalan yang kuat, atau kemampuan yang mengagumkan dalam berceramah dan menyampaikan materi kajian, atau gelar dan titel yang disandang,

tanpa adanya wujud nyata dan pengaruh dari kemanfaatan ilmu tersebut bagi orang yang mempelajarinya dan menggunakannya. Abdullah bin Mas'ud yang berkata: “Bukanlah ilmu itu (hanya) dengan banyak (menghafal) hadits, akan

{إنما يخشى الله من عباده العلماء إن الله عزيز غفور}

tetapi ilmu (yang bermanfaat) itu (timbul) dari besarnya rasa takut (kepada Allah)

"Sesungguhnya yang memiliki rasa takut kepada Allah diantara hamba-hambanya hanyalah orang-orang yang berilmu (tentang agama Allah). Sesungguhnya Allah Maha Perkasa lagi Maha Pengampun" (QS Faathir:28).

Adapun dalam kitab “Al Khusyuu’ fish shalaah” (hal. 16) Imam Ibnu Rajab Al Hambali berkata: “Ilmu yang bermanfaat adalah ilmu yang masuk (dan menetap) ke dalam relung hati (manusia), yang kemudian melahirkan rasa tenang, takut, tunduk, merendahkan dan mengakui kelemahan diri di hadapan Allah. Dengan dasar diatas, penelitian kecil ini bertujuan untuk memberikan kontribusi terhadap ilmu pengetahuan dan juga memberikan manfaat bagi dunia teknologi dalam bidang pemetaan. (Ilmu yang Bermanfaat, Abdullah Taslim:2)

Berkaitan dengan fungsi navigasi dan pemetaan Pada surah al-Kahf (18) ayat ke 17, berfirman,

❖ وَتَرَى الشَّمْسَ إِذَا طَلَعَتْ تَزَّوُّرُ عَنْ كَهْفِهِمْ ذَاتَ الْيَمِينِ وَإِذَا غَرَبَتْ
تَقَرَّبُ إِلَيْهِمْ ذَاتَ الشِّمَالِ وَهُمْ فِي فَجْوَةٍ مِّنْهُ ذَلِكَ مِنْ آيَاتِ اللَّهِ مَن
يَهْدِ اللَّهُ فَهُوَ الْمُهْتَدِ ۖ وَمَن يُضِلِلْ فَلَن تَجِدَ لَهُ وَلِيًّا مُّرْشِدًا ﴿١٧﴾

"Dan engkau akan melihat matahari ketika terbit, cenderung ke kanan dari gua mereka; dan apabila ia terbenam, meninggalkan mereka ke arah kiri, sedang mereka berada dalam satu lapangan gua itu. Yang demikian ialah dari tanda-tanda (yang membuktikan kekuasaan) Allah. Sesiapa yang diberi hidayah petunjuk oleh Allah, maka dia lah yang berjaya mencapai kebahagiaan; dan sesiapa yang disesatkanNya maka engkau tidak sekali-kali akan beroleh sebarang penolong yang dapat menunjukkan (jalan yang benar) kepadanya."

Dari tafsir Jalaludin As-Suyuthi dan Jalaludin Muhammad Ibnu Ahmad Al-Mahally dijelaskan, (Dan kamu akan melihat matahari ketika terbit, condong) Lafal Tazaawaru dapat dibaca dengan memakai Tasydid atau Takhfif, artinya melenceng (dari gua mereka ke sebelah kanan) ke arah sebelah kanan (dan bila matahari itu terbenam menjauhi mereka ke sebelah kiri) yakni membiarkan mereka dan melewati mereka, hingga sinar matahari sama sekali tidak mengenai mereka (sedangkan mereka berada di tempat yang luas dalam gua itu) yakni gua yang luas, sehingga mereka selalu mendapatkan tiupan angin yang segar lagi menyejukkan. (Itu) yakni hal yang telah disebutkan (adalah sebagian tanda-tanda Allah) bukti-bukti yang menunjukkan akan kekuasaan-Nya. (Barang siapa yang diberi petunjuk oleh Allah, maka dialah yang mendapat petunjuk dan barang siapa yang disesatkan-Nya, maka kamu tak akan mendapatkan seorang pemimpin pun yang dapat memberi petunjuk kepadanya) (Tafsir Jalalain : 179)

Al-Quran adalah petunjuk hanya bagi orang yang takwa. Setiap manusia harus memiliki panduan dalam hidup agar dapat memahami dan memperoleh pelajaran dari padanya. Sebuah Sistem petunjuk atau sistem navigasi dalam dunia

IT sangat dibutuhkan untuk mempermudah manusia menentukan jalan yang tepat. Sistem navigasi membuat perjalanan menjadi tenang karena kita tahu kemana tujuan kita, begitu juga Al-Qur'an, adalah sebagai petunjuk untuk manusia yang bertakwa hidup di dunia. (Mukjizat Al-Qur'an dan As-Sunnah tentang IPTEK : 96)



1.2 Rumusan Masalah

Berdasarkan uraian permasalahan diatas, maka dapat dirumuskan permasalahannya adalah bagaimana membuat suatu aplikasi yang dapat membantu manusia untuk mengamati daerah yang sulit dijangkau atau bahkan berbahaya dengan bantuan sensor-sensor robot sehingga mengurangi resiko yang akan dialami saat proses observasi suatu lingkungan.

1.3 Batasan Masalah

Dalam penelitian ini penulis membatasi masalah pada pembuatan aplikasi generate map.

1. menggunakan metode EKF-SLAM berdasarkan data yang diperoleh dari sensor yang dikirim oleh simulator mikrokontroler arduino pada perangkat mobile platform android.
2. Batasan dari lingkungan yang sulit dijangkau adalah daerah reruntuhan pasca bencana atau goa-goa yang nanti akan disimulasikan menggunakan ruang buatan.
3. Aplikasi tidak berjalan di multiplatform namun hanya dibatasi pada platform android saja untuk mempermudah proses pembuatan aplikasi.
4. Data dikirim melalui mikrokontroler arduino. Aplikasi ini akan mengolah data masukan dari mikrokontroler tersebut untuk diproses menjadi output peta simulasi.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan aplikasi yang dapat membangun jalur dan lingkungan gerak robot berdasarkan data mikrokontroller sebagai simulasi dari robot.

1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi *generta map* ini adalah untuk mempermudah proses observasi suatu lingkungan yang sulit dijangkau oleh manusia dengan memanfaatkan sebuah robot autonomous untuk menjangkau daerah tersebut.

1.6 Sistematika Penulisan

Penulisan skripsi ini tersusun dalam 5 (lima) bab dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pendahuluan, berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penyusunan skripsi.

BAB II KAJIAN PUSTAKA

Kajian pustaka, berisi tentang teori-teori dan referensi yang melandasi penyusunan skripsi.

BAB III PERANCANGAN SISTEM

Menganalisa kebutuhan sistem untuk membuat *aplikasi* meliputi spesifikasi kebutuhan *software* dan langkah-langkah pembuatan *aplikasi*.

BAB IV HASIL DAN PEMBAHASAN

Menjelaskan tentang pengujian *metode* yang telah diterapkan pada pembuatan aplikasi

BAB V PENUTUP

Berisi kesimpulan dan saran mengenai hasil penelitian dari aplikasi generate map menggunakan metode EKF-SLAM ini.





BAB II

KAJIAN PUSTAKA

2.1 Dalam Pandangan Islam

Pada surah al-Kahf (18) ayat ke 17, berfirman,

﴿ وَتَرَى الشَّمْسَ إِذَا طَلَعَتْ تَزَّوُّرًا عَنْ كَهْفِهِمْ ذَاتَ الْيَمِينِ وَإِذَا غَرَبَتْ تَقَرَّبُ إِلَيْهِمْ ذَاتَ الشِّمَالِ وَهُمْ فِي فَجْوَةٍ مِنْهُ ذَلِكَ مِنْ آيَاتِ اللَّهِ لِيَهْدِيَ اللَّهُ لِمَنْ يَشَاءُ فَمَا تَتَجَدَّدُ لَهُمْ وَلِيَأْمُرُوا بِالسَّعْيِ وَالْإِحْسَانِ ﴾

Yang artinya: "Dan engkau akan melihat matahari ketika terbit, cenderung ke kanan dari gua mereka; dan apabila ia terbenam, meninggalkan mereka ke arah kiri, sedang mereka berada dalam satu lapangan gua itu. Yang demikian ialah dari tanda-tanda (yang membuktikan kekuasaan) Allah. Sesiapa yang diberi hidayah petunjuk oleh Allah, maka dia lah yang berjaya mencapai kebahagiaan; dan sesiapa yang disesatkanNya maka engkau tidak sekali-kali akan beroleh sebarang penolong yang dapat menunjukkan (jalan yang benar) kepadanya."

Dari tafsir Jalaludin As-Suyuthi dan Jalaludin Muhammad Ibnu Ahmad Al-Mahally menyebutkan, (Dan kamu akan melihat matahari ketika terbit, condong) Lafal Tazaawaru dapat dibaca dengan memakai Tasydid atau Takhfif, artinya melenceng (dari gua mereka ke sebelah kanan) ke arah sebelah kanan (dan bila matahari itu terbenam menjauhi mereka ke sebelah kiri) yakni membiarkan mereka dan melewati mereka, hingga sinar matahari sama sekali tidak mengenai

mereka (sedangkan mereka berada di tempat yang luas dalam gua itu) yakni gua yang luas, sehingga mereka selalu mendapatkan tiupan angin yang segar lagi menyejukan. (Itu) yakni hal yang telah disebutkan (adalah sebagian tanda-tanda Allah) bukti-bukti yang menunjukkan akan kekuasaan-Nya. (Barang siapa yang diberi petunjuk oleh Allah, maka dialah yang mendapat petunjuk dan barang siapa yang disesatkan-Nya, maka kamu tak akan mendapatkan seorang pemimpin pun yang dapat memberi petunjuk kepadanya) (Tafsir Jalalain : 179)

Al-Quran adalah petunjuk hanya bagi orang yang takwa. Setiap manusia harus memiliki panduan dalam hidup agar dapat memahami dan memperoleh pelajaran dari padanya. Sebuah Sistem petunjuk atau sistem navigasi dalam dunia IT sangat dibutuhkan untuk mempermudah manusia menentukan rute yang tepat. Sistem navigasi membuat perjalanan menjadi tenang karena kita tahu kemana tujuan kita, begitu juga Al-Qur'an, adalah sebagai petunjuk untuk manusia yang bertakwa hidup di dunia.

2.2 Simultaneous localization and Mapping (SLAM)

Simultaneous localization and Mapping (SLAM) adalah teknik yang digunakan untuk membangun sebuah peta pada lingkungan yang belum diketahui melalui bantuan robot ataupun *autonomous vehicle*, atau juga mengupdate peta yang sudah diketahui dan pada waktu yang sama mengenali lokasi dari robot tersebut. Buku "SLAM for Dummies" menjelaskan bahwa SLAM bukan merupakan sebuah algoritma tunggal, melainkan sebuah konsep utuh yang

didalamnya memuat beberapa algoritma yang bekerja secara bersama-sama pada kondisi-kondisi tertentu. (SLAM for Dummies 2005 : 6).

Banyak penelitian yang telah mempelajari SLAM dalam dua dekade terakhir. Riisgaard dan Blas (2003) memberikan panduan pemula untuk masalah SLAM dan memberikan gambaran yang luas dari proses SLAM. Para penulis memberikan gambaran singkat dan sejarah SLAM. Makalah ini membahas jenis perangkat keras yang dibutuhkan untuk memecahkan SLAM termasuk sensor sonar, sensor laser, dan sensor visi, dan keuntungan dan kerugian masing-masing. Para penulis daftar tiga bagian utama dari SLAM sebagai: ekstraksi landmark, asosiasi data, dan Extended Kalman Filter. Makalah ini juga berisi informasi dan analisis mengenai dua metode ekstraksi fitur dan gambaran EKF SLAM yang menyediakan dasar pemahaman untuk EKF SLAM. Kelebihan dari penelitian yang dikerjakan oleh Riisgaard dan Blas adalah meeka berhasil mengimplementasikan EKF-SLAM dengan akurasi yang baik pada nilai *random* dan lebih akurat bila disimulasikan pada map yang relatif kecil, namun untuk *map* yang besar waktu yang dibutuhkan tidak optimal. (An Analysis of Simultaneous Localization and Mapping (SLAM) Algorithms 2013 : 11).

2.3 Kompleksitas SLAM

Penelitian dan Ahli pada bidang Artificial Intelegence berusaha untuk memecahkan masalah pada SLAM. SLAM membutuhkan komputasi yang kuat untuk mencakupi sebidang area dan memproses hasil data untuk menampilkan peta dan lokasi. Pada tahun 2008 topik rangkuman tentang SLAM adalah salah

satu hal yang menjadi tantangan tersendiri dalam dunia robotika, namun semua pendekatan selama ini belum dapat menangani lingkungan yang luas, dikarenakan biaya komputasi dan lingkungan yang tidak menentu yang kemudian menjadi penghalang ketika skenario ruang menjadi besar.

Secara umum, solusi 3D lengkap SLAM adalah proses komputasi yang sangat intensif karena menggunakan *real-time particle filters*, *sub-mapping strategies* atau kombinasi metode lain. Robot yang menggunakan *embedded system* tidak dapat mengimplementasikan metode SLAM secara penuh dikarenakan sumber daya komputasi yang terbatas. Nguyen V., Harati A., & Siegwart R. (2007) telah berkontribusi dalam *embedded robotics* dengan cepat, solusi ringan yang dinamakan Ort hoSLAM, yang mematahkan kompleksitas dari lingkungan kedalam bidang *orthogonal*.

Dengan hanya memetakan bidang yang orthogonal satu dengan yang lainnya, struktur kebanyakan dari lingkungan *indoor* dapat diestimasi cukup akurat. Algoritma OrthoSLMA mengurangi SLAM ke masalah estimasi linier sejak hanya single line di proses setiap waktu.

2.4 Metode dalam SLAM

Terdapat beberapa pendekatan dalam usaha membangun peta dan lokalisasi robot dengan kekurangan dan kelebihan tersendiri. Beberapa metode

tersebut adalah FastSLAM 1.0, FastSLAM 2.0, L-SLAM, GraphSLAM, Occupancy Grid SLAM, DP-SLAM, LSD-SLAM.

2.5 Problem Teknis SLAM

SLAM dapat dikatakan sebagai masalah Chicken or Egg: Sebuah peta yang tidak bias dibutuhkan untuk proses lokalisasi ketika posisi akurat dibutuhkan untuk membangun peta tersebut. Disamping itu, jawaban dari dua karakteristik pertanyaan tidak serta merta sehingga mungkin terdengar tidak pasti dalam membedakan pergerakan robot dari beberapa sensornya. Secara umum, dikarenakan biaya dari gangguan dari lingkungan teknis, SLAM tidak dilayani dengan hanya sebuah solusi yang ringkas, tetapi dengan banyak *physical concepts* untuk membantu hasilnya. Jika perulangan selanjutnya dari peta dari hasil pengukuran dan arah yang telah dilalui .

Jika pada iterasi berikutnya dari peta membangun jarak yang diukur dan arah memiliki nilai yang tidak akurat, maka seluruh *features* dari lingkungan yang ditambahkan kedalam peta akan mengandung error. Seiring waktu dan gerakan robot, *locating* dan pemetaan kesalahan dibangun secara kumulatif, terlalu mendistorsi peta sehingga kemampuan robot untuk menentukan lokasi sebenarnya dan menuju dengan akurasi yang memadai.

Terdapat beberapa jenis teknik untuk mengimbangi error, seperti mengenali *feature* pada peta misalnya data *association* atau pengulangan closure detection, dan mencondongkan ulang bagian terakhir dari peta untuk memastikan

dua contoh dari fitur yang menjadi salah satu. Teknik statistik yang digunakan dalam SLAM termasuk Kalman filter, filter partikel (Monte Carlo metode) dan pencocokan memindai data jangkauan. Mereka memberikan estimasi fungsi probabilitas posterior untuk pose robot dan untuk parameter peta. Teknik set-anggota terutama didasarkan pada propagasi kendala interval. Mereka menyediakan satu set yang membungkus pose robot dan pendekatan set peta.

2.5.1 Mapping

SLAM dalam komunitas mobile robotika umumnya mengacu pada proses menciptakan peta geometris yang konsisten dari lingkungan. Peta Topological adalah metode representasi lingkungan yang menangkap konektivitas (yaitu, topologi) dari lingkungan daripada menciptakan peta geometris yang akurat. Pendekatan SLAM Topological telah digunakan untuk menegakkan konsistensi global dalam algoritma SLAM metrik. SLAM disesuaikan dengan sumber daya yang tersedia, maka tidak ditujukan untuk kesempurnaan, namun harus sesuai pada kepatuhan operasional. Pendekatan yang ada bekerja di kendaraan udara tak berawak, kendaraan bawah air otonom, penemu planet, yang baru muncul robot domestik dan bahkan di dalam tubuh manusia.

Hal ini pada umumnya dianggap sebagai "pemecahan" masalah SLAM telah menjadi salah satu prestasi penting dari penelitian robotika dalam dekade terakhir. Masalah terkait asosiasi data dan kompleksitas komputasi adalah salah satu masalah belum sepenuhnya diselesaikan.

Kemajuan baru yang signifikan dalam fitur berbasis SLAM literatur melibatkan pemeriksaan ulang dasar probabilistik untuk Simultan Lokalisasi dan Pemetaan (SLAM) di mana ia berpose dalam hal multi-objek Bayesian filtering dengan set terbatas acak yang memberikan kinerja yang unggul untuk memimpin algoritma berbasis fitur SLAM dalam skenario pengukuran menantang dengan tingkat akurat yang tinggi dan tingkat deteksi terjawab tinggi tanpa perlu asosiasi data.

2.5.2 Sensing

SLAM akan selalu menggunakan beberapa jenis sensor untuk memperoleh data dengan kesalahan statistik pada setiap sensornya. Sensor statistik adalah persyaratan wajib untuk mengatasi metrik bias dan dengan kebisingan dalam langkah-langkah. Berbagai jenis sensor menimbulkan algoritma SLAM yang berbeda yang asumsi yang paling sesuai dengan sensor. Sebagai contoh, scan laser atau fitur individual memberikan rincian dari banyak poin besar dalam suatu daerah, tetapi berbasis sensor sentuh relatif jarang mengharuskan prior kuat untuk mengimbangnya.

Sensor optik satu-dimensi (balok tunggal) atau 2D-(sweep) rangefinders laser, LIDAR 3D High Definition, 3D Flash LIDAR, 2D atau 3D sensor sonar dan satu atau lebih kamera 2D. Sejak tahun 2005, telah ada penelitian intensif VSLAM (SLAM visual) menggunakan terutama visual (kamera) sensor, karena di mana-mana peningkatan kamera seperti yang ada di perangkat mobile. bentuk terbaru lainnya dari SLAM

mencakup SLAM taktil (penginderaan dengan sentuhan lokal saja), radar SLAM, dan wifi-SLAM (penginderaan oleh kekuatan dekatnya titik akses wifi).

Pendekatan baru-baru ini menerapkan kuasi-optik nirkabel mulai untuk multi-lateration (RTLS) atau multi-angulasi dalam hubungannya dengan SLAM sebagai penghargaan untuk tindakan nirkabel tidak menentu. Khusus jenis SLAM untuk pejalan kaki manusia menggunakan unit pengukuran inersia sepatu dipasang sebagai sensor utama dan bergantung pada kenyataan bahwa pejalan kaki dapat menghindari dinding. Pendekatan ini disebut FootSLAM dapat digunakan untuk secara otomatis membangun denah bangunan yang kemudian dapat digunakan oleh positioning system dalam ruangan.

2.5.3 Locating

Hasil dari penginderaan akan memberi masukan data kedalam algoritma untuk mencari. Menurut proposisi geometri, penginderaan apapun harus menyertakan setidaknya satu literasi dan $(n + 1)$ menentukan persamaan untuk masalah- n dimensi. Selain itu, harus ada beberapa tambahan pengetahuan apriori tentang orientasi hasil terhadap sistem absolut atau relatif koordinat dengan rotasi dan mirroring.

2.5.3 Modeling

Kontribusi untuk pemetaan dapat bekerja dalam pemodelan 2D dan perwakilan masing-masing atau di 3D modeling dan representasi proyektif 2D juga. Sebagai bagian dari model, kinematika robot disertakan, untuk

meningkatkan perkiraan merasakan dalam kondisi yang melekat dan ambient kebisingan. Model dinamis menyeimbangkan kontribusi dari berbagai sensor, berbagai model kesalahan parsial dan akhirnya terdiri dalam penggambaran maya tajam sebagai peta dengan lokasi dan judul robot karena beberapa awan probabilitas. Pemetaan adalah menggambarkan akhir dari model seperti, peta adalah salah penggambaran tersebut atau istilah abstrak untuk model.

2.5.4 Exploration

"Active SLAM" mempelajari masalah gabungan SLAM dengan memutuskan tempat untuk memindahkan berikutnya dalam rangka membangun peta seefisien mungkin. Kebutuhan untuk eksplorasi aktif terutama diucapkan dalam rezim penginderaan jarang seperti taktil SLAM. SLAM aktif umumnya dilakukan dengan mendekati entropi dari peta di bawah tindakan hipotesis. "Multi agen SLAM" meluas masalah ini dengan kasus beberapa robot koordinasi diri untuk mengeksplorasi secara optimal.

2.6 Extended Kalman Filter (EKF-SLAM)

Tim Bailey, Juan Nieto (2002). Pada paper Consistency of the EKF-SLAM Algorithm ini, mereka menunjukkan simulasi EKF-SLAM untuk *mapping*. Mereka menunjukkan bahwa algoritma akan selalu menunjukkan ketidak konsistenan. Kondisi tersebut sulit dideteksi bila tidak mengetahui *ground-truth* dari *vehicle*. Solusi konvensional seperti menambahkan penstabil noise pada landmark, menggunakan filter, tidak meningkatkan situasi.

Ketidakkonsistenan akan selalu ada. Namun kelebihan dari paper ini adalah mereka dapat memberikan simulasi EKF-SLAM dengan baik dan jelas.

Durrant-Whyte dan Bailey (2006) pada dua makalah dimaksudkan sebagai tutorial untuk memahami masalah SLAM. Paper ini diawali dengan memperkenalkan masalah SLAM. Paper ini juga masuk ke dalam sejarah dan perkembangan masalah SLAM, mendeskripsikan asumsi awal tentang solusi dan terobosan di suatu area. Para penulis menjelaskan bahwa korelasi di lokasi landmark harus digunakan untuk menentukan lokasi landmark bukannya dibuang sebagai sebuah *noise*. Artikel ini masuk ke detail tentang perhitungan terlibat dalam SLAM probabilistik, yang melibatkan menggunakan Teorema Bayes dan model Markov untuk menghitung probabilitas pengamatan landmark dan lokasi kendaraan menggunakan data sebelumnya dari lokasi kendaraan, pengamatan landmark, input kontrol, dan landmark set. Tutorial ini juga menyebutkan beberapa implementasi dari SLAM serta sumber perangkat lunak open source untuk simulasi SLAM.

2.7 Algoritma EKF-SLAM

Dari konsep SLAM diatas terdapat metode yang dapat membantu optimalisasi dalam membangun peta yaitu Extended Kalman Filter (EKF-SLAM). Kalman Filter adalah bentuk algoritma untuk *predictor-corrector* yang digunakan secara luas ekstensif pada sistem kontrol untuk memperkirakan lokasi yang sulit terukur.

SLAM ditentukan oleh posisi dari *position and heading* dari robot dan juga dari lokasi *landmark* pada sebuah lingkungan yang diamati. Kadaan pada waktu k diwakili oleh oleh state-vector \mathbf{X}_k .

$$\mathbf{x}_k = [x_{v_k}, y_{v_k}, \phi_{v_k}, x_1, y_1, \dots, x_N, y_N]^T = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{m} \end{bmatrix} \quad (1)$$

Diketahui bahwa parameter parameter pada peta yaitu $\mathbf{m} = [x_1, y_1, \dots, x_n, y_n]^T$ tidak memiliki ketergantungan terhadap waktu sebab peta dimodelkan sebagai lingkungan stationary atau tidak bergerak.

Untuk mendeskripsikan pergerakan dari robot digunakan model kinematic untuk trajectory dari roda depan robot pada gerakan putar yang konstan (diasumsikan bahwa perputaran dari roda adalah 0 slip).

$$\mathbf{x}_{v_k} = \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix} \quad (2)$$

Disini saat waktu dari $k - 1$ menuju k ditunjukkan dengan ΔT , dan selama periode kecepatan V_k dan kendali sudut γ_k dari pergerakan roda depan robot diasumsikan konstan. Secara bersama-sama, nilai untuk kecepatan dan kendali $\mathbf{u}_k = [V_k, \gamma_k]^T$ adalah variable kontrol. Jarak roda antara bagian depan sampai belakang adalah B . Hubungan antara pergerakan robot dan landmark pada model SLAM secara singkat adalah;

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{m} \end{bmatrix} \quad (3)$$

Untuk pengukuran antara robot dan landmark $\mathbf{m}_i = [\mathbf{X}_i, \mathbf{Y}_i]^T$, model untuk observasinya didapat melalui;

$$\mathbf{z}_{i_k} = \mathbf{h}_i(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(x_i - x_{v_k})^2 + (y_i - y_{v_k})^2} \\ \arctan \frac{y_i - y_{v_k}}{x_i - x_{v_k}} - \phi_{v_k} \end{bmatrix} \quad (4)$$

Pergerakan dari robot, observasi, dan pengukuran dari parameter \mathbf{U} tidaklah eksak, namun masih terdapat banyak noise yang masih dibutuhkan banyak proses estimasi.

2.8 Aplikasi Hybrid

Menurut Hezdy Muhammad Aplikasi ini dikembangkan dalam bahasa yang sejenis dengan aplikasi web, namun aplikasi hybrid memiliki platform native sendiri tidak bergantung pada browser. (platform native yang digunakan oleh aplikasi hybrid pada dasarnya merupakan browser yang dihilangkan beberapa fiturnya. Atau bisa dibilang browser yang memiliki tujuan khusus hanya untuk aplikasi hybrid tersebut. Aplikasi hybrid merupakan aplikasi web yang berjalan di device. Dalam penelitian yang saya lakukan, aplikasi *hybird* disini menggunakan platform pendukung yaitu; Cordova phonegap dan Processing.

2.9 Phonegap Cordova

Phonegap adalah framework *opensource* yang dapat membuat aplikasi mobile menggunakan standar web API untuk multi *platform*. Phonegap dapat membangun aplikasi mobile dengan menggunakan teknologi web seperti HTML, CSS, dan JavaScript.

Inti dari aplikasi PhoneGap menggunakan HTML5 dan CSS3 untuk render mereka, dan JavaScript untuk logika mereka. Meskipun HTML5 sekarang menyediakan akses ke perangkat keras yang mendasari seperti accelerometer, kamera dan GPS, dukungan browser untuk akses perangkat berbasis HTML5 tidak konsisten di seluruh browser mobile, khususnya versi Android. Untuk mengatasi keterbatasan ini, kerangka PhoneGap embeds kode HTML5 di dalam WebView asli pada perangkat, menggunakan antarmuka fungsi asing untuk mengakses sumber-sumber asli perangkat.

PhoneGap juga mampu diperpanjang dengan plug-in asli yang memungkinkan pengembang untuk menambahkan fungsionalitas yang dapat dipanggil dari JavaScript, yang memungkinkan untuk komunikasi langsung antara lapisan asli, dan halaman HTML5. PhoneGap termasuk plugin dasar yang memungkinkan akses ke accelerometer perangkat, kamera, mikrofon, kompas, sistem file, dan banyak lagi.

Namun, penggunaan teknologi berbasis web menyebabkan banyak aplikasi PhoneGap untuk berjalan lebih lambat dari aplikasi asli dengan fungsi serupa. [25] Adobe Systems memperingatkan bahwa aplikasi yang dibangun menggunakan PhoneGap dapat ditolak oleh Apple karena terlalu lambat atau tidak merasa "asli"

Cukup (memiliki penampilan dan fungsi sesuai dengan apa yang pengguna telah datang ke harapkan pada platform).

PhoneGap saat ini mendukung pengembangan untuk sistem operasi Apple iOS, BlackBerry, Google Android, LG webOS, Microsoft Windows Phone (7 dan 8), Nokia Symbian OS, Tizen (SDK 2.x), Bada, Firefox OS, dan Ubuntu Touch. Tabel di bawah ini adalah daftar fitur yang didukung untuk sistem operasi android.

Tabel 2.1 Phonegap Suported Platform

Feature	Android 1.0 - 4.4
Accelerometer	Yes
Camera	Yes
Compass	Yes
Contacts	Yes
File	Yes
Geolocation	Yes
Media	Yes
Network	Yes
Notification (alert, sound, vibration)	Yes
Storage	Yes

2.10 Processing

Processing adalah bahasa pemrograman, komunitas online, dan *development enviroment*. Sejak 2001, processing adalah bahasa pemrograman yang dipromosikan untuk menangani seni visual dan literatur visual teknologi. Processing dibuat untuk memahami bahasa komputer secara fundamental dengan menggunakan konteks visual.

Processing termasuk sketsa, alternatif minimal untuk lingkungan pengembangan terintegrasi (IDE) untuk membuat proyek. Setiap sketsa Processing sebenarnya adalah subclass dari kelas PApplet Java yang mengimplementasikan sebagian besar fitur bahasa Pengolahan ini.

Ketika pemrograman dalam Pengolahan, semua kelas tambahan didefinisikan akan diperlakukan sebagai kelas batin ketika kode tersebut diterjemahkan ke Jawa murni sebelum kompilasi. Ini berarti bahwa penggunaan variabel statis dan metode dalam kelas dilarang kecuali Anda secara eksplisit memberitahu Pengolahan yang ingin Anda kode dalam mode Jawa murni.

Processing juga memungkinkan bagi pengguna untuk membuat kelas mereka sendiri dalam sketsa PApplet. Hal ini memungkinkan untuk tipe data yang kompleks yang dapat mencakup sejumlah argumen dan menghindari keterbatasan hanya menggunakan tipe data standar seperti: int (integer), char (karakter), float (bilangan real), dan warna (RGB, ARGB, hex).

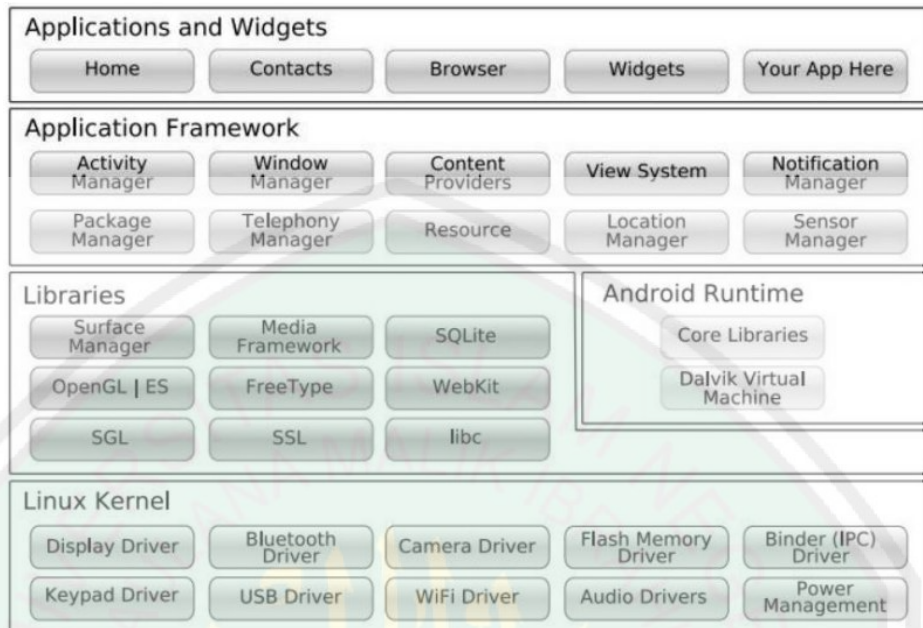
2.11 Android

Pengertian Android menurut Sifaat (2001:1) mengemukakan bahwa,

“android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android diakuisisi oleh Google pada Juli 2005, dan baru dirilis perdana pada 5 November 2007. Android berlisensi di bawah GNU, General Public Lisensi Versi 2 (GPLv2), yang memperbolehkan pihak ketiga untuk mengembangkannya dengan menyertakan term yang sama. Pendistribusiannya di bawah Lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya”.

Android dirancang dengan arsitektur sebagai berikut (Safaat, 2001:6-9) dan akan ditunjukkan dengan gambar 2.1:

1. *Application dan Widgets*
2. *Application Frameworks*
3. *Libraries*
4. *Android Run Time*
5. *Linux Kernel*



Gambar 2.1 Arsitektur Android (Sumber: Safaat, 2001:9)

Beberapa keunggulan Platform Android menurut Safaat (2001:3) adalah sebagai berikut (Safaat, 2001:3):

1. Lengkap (*Complete Platform*).
2. Terbuka (*Open Source Platform*).
3. Bebas (*Free Platform*).

2.12 Bahasa Pemrograman Java

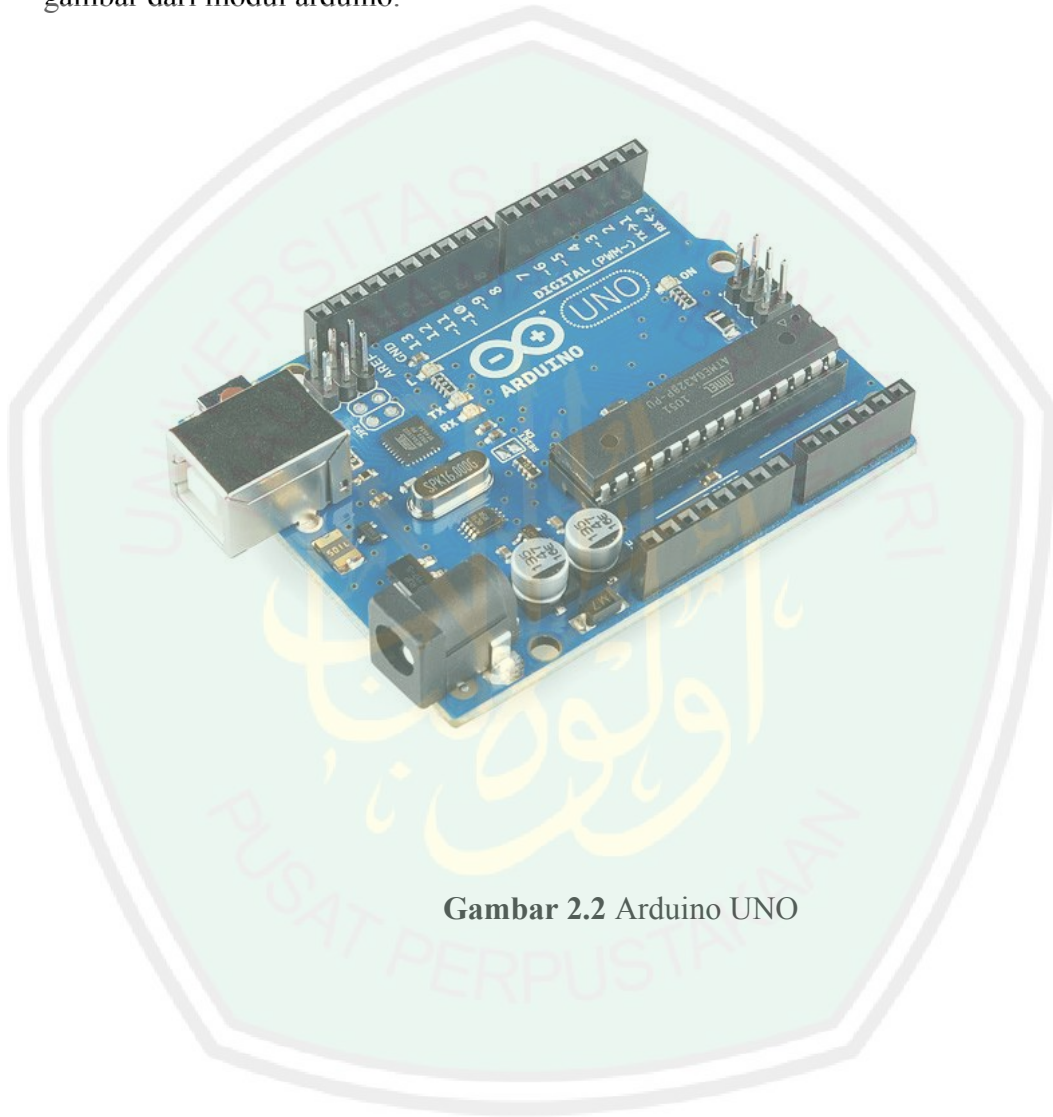
Java adalah bahasa pemrograman yang multi platform dan multi device. Sekali anda menuliskan sebuah program dengan menggunakan Java, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support Java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis Java ini dikompulasikan ke dalam p-code dan bisa dijalankan dengan Java Virtual Machine. Fungsionalitas dari Java ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik.

2.13 Arduino

Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri. Proyek ini berawal di Ivrea, Italia pada tahun 2005. Sekarang telah lebih dari 120.000 unit terjual. Pendirinya adalah Massimo Banzi dan David Cuartielles. Karena rancangan hardware dan software Arduino bersifat open-source, produsen lain bebas untuk menirunya, misalnya:

- Freduino
- Cosmo Black Star
- Freduino MaxSerial
- Zigduino

Tim pengembang Arduino adalah Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis, dan Nicholas Zambetti. Berikut ini adalah gambar dari modul arduino.



Gambar 2.2 Arduino UNO

2.14 Bluetooth

Bluetooth adalah spesifikasi industri untuk jaringan kawasan pribadi (personal area networks atau PAN) tanpa kabel. Bluetooth menghubungkan dan dapat dipakai untuk melakukan tukar-menukar informasi di antara peralatan-peralatan. Spesifikasi dari peralatan Bluetooth ini dikembangkan dan didistribusikan oleh kelompok Bluetooth Special Interest Group. Bluetooth beroperasi dalam pita frekuensi 2,4 Ghz dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real time antara host-host bluetooth dengan jarak terbatas. Kelemahan teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah.

Awal mula dari Bluetooth adalah sebagai teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas yakni sekitar 10 meter. Bluetooth berupa kartu yang menggunakan frekuensi radio standar IEEE 802.11 dengan jarak layanan yang terbatas dan kemampuan data transfer lebih rendah dari kartu untuk Wireless Local Area Network (WLAN).

Pembentukan Bluetooth dipromotori oleh 5 perusahaan besar Ericsson, IBM, Intel, Nokia dan Toshiba membentuk sebuah Special Interest Group (SIG) yang meluncurkan proyek ini. Pada bulan Juli 1999 dokumen spesifikasi

bluetooth versi 1.0 mulai diluncurkan. Pada bulan Desember 1999 dimulai lagi pembuatan dokumen spesifikasi bluetooth versi 2.0 dengan tambahan 4 promotor baru yaitu 3Com, Lucent Technologies, Microsoft dan Motorola. Saat ini, lebih dari 1800 perusahaan di berbagai bidang bergabung dalam sebuah konsorsium sebagai adopter teknologi bluetooth. Walaupun standar Bluetooth SIG saat ini 'dimiliki' oleh grup promotor tetapi ia diharapkan akan menjadi sebuah standar IEEE (802.15).

Dalam penelitian ini module yang digunakan bersama dengan arduino adalah module bluetooth DFRobot Bluetooth V3. Seperti gambar dibawah ini.

Degan Spesifikasi :

- The Bluetooth chip: CSR BC417143
- Bluetooth protocol: Bluetooth Specification v2.0 + EDR
- USB Protocol: USB v1.1/2.0
- Operating frequency: 2.4 ~ 2.48GHz unlicensed ISM band
- Modulation: GFSK (Gaussian Frequency Shift Keying)
- Transmit Power: $\leq 4\text{dBm}$, Class 2
- Transmission distance: 20 ~ 30m in free space
- Sensitivity: $\leq -84\text{dBm}$ at 0.1% BER
- Transfer rate: Asynchronous: 2.1Mbps (Max) / 160 kbps; Synchronous: 1Mbps/1Mbps
- Safety features: Authentication and encryption
- Support profiles: Bluetooth serial port
- Serial port baud rate: 4800 ~ 1382400 / N / 8 / 1 default: 9600

- LED indicator: STATE state:
- Input Voltage: +3.5 V ~ +8 V DC and 3.3V DC/50mA
- Working temperature: -20 × ~ +55 ×
- Module Size: 40 ,, 20 ,, 13mm



Gambar 2.3 Bluetooth Module

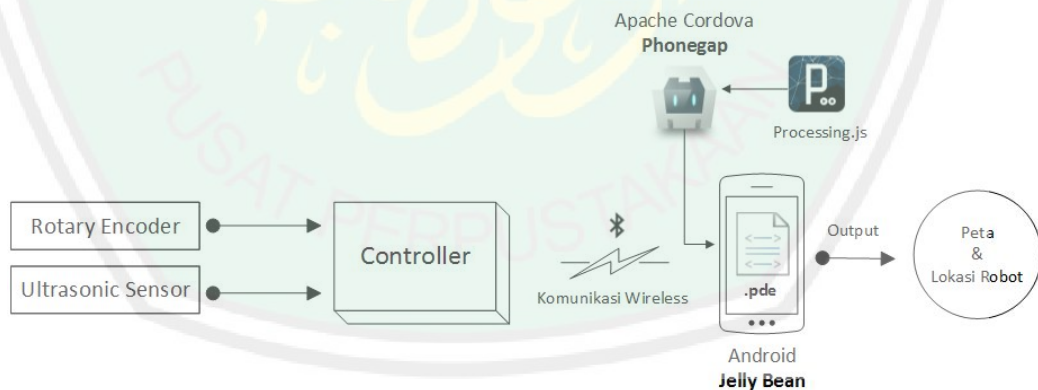


BAB III PERANCANGAN SISTEM

3.1 Analisa dan Perancangan Sistem

Beberapa konsep yang ada saat ini masih belum memanfaatkan perangkat mobile. Perkembangan perangkat *mobile* yang semakin canggih dapat dimanfaatkan sebagai resoure yang baik untuk memproses jalur gerak robot menggunakan metode EKF-SLAM ini. Perangkat mobile android diatas v2.x *support canvas* yang bisa digunakan untuk menghasilkan output 2D.

Dari fitur tersebut rancangan konsep yang akan dibuat oleh peneliti adalah menghasilkan jalur gerak robot dan posisi robot pada perangkat mobile. Secara umum, rancangan sistem yang digunakan seperti tampak pada gambar 3.1



Gambar 3.1 Rancangan Sistem Aplikasi

Dari diagram sistem diatas terdapat beberapa sensor, fitur, dan platform yang memiliki fungsi berbeda-beda.

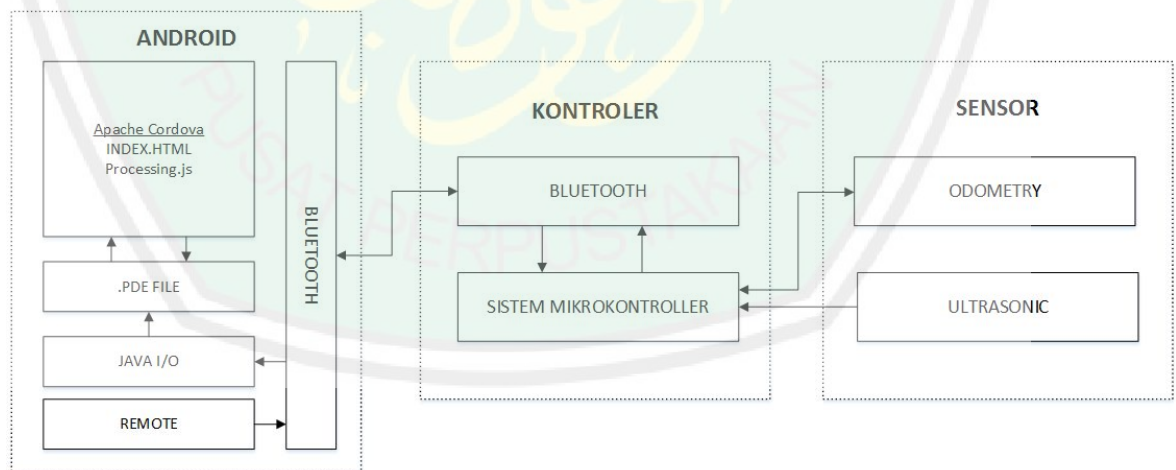
1. **Mikrokontroller**, digunakan untuk mengontrol rangkaian elektronik seperti sensor ultrasonic dan rotary encoder dan juga digunakan untuk menyimpan program.
2. **Rotary Encoder**, digunakan untuk menentukan arah hadap robot dan jarak yang ditempuh oleh robot.
3. **Sensor Ultrasonic**, digunakan untuk mengetahui jarak halangan terhadap robot pada lingkungan yang diobeservasi
4. **Komunikasi wireless**, digunakan untuk mengirimkan data sensor ke perangkat android untuk selanjutnya diolah menjadi bentuk jalur gerak dari simulasi robot dan posisinya.
5. **Platform Apache Cordova**, digunakan untuk menampung aplikasi dalam bentuk *webbased application* pada perangkat android.
6. **Platform Processing.js**, digunakan untuk menghasilkan output dalam bentuk animasi.
7. **Perangkat Android**, menjalankan aplikasi dan menampilkan output pada layar perangkat android

Keuntungan dari menggunakan perangkat android adalah pengamat dapat mendapatkan data secara *mobile* dan *realtime*.

3.1.1 Prinsip Kerja Sistem

Secara umum terdapat 3 diagram blok untuk menggambarkan prinsip kerja sistem keseluruhan. Pertama-tama Aplikasi android yang dijalankan akan meminta untuk menjalankan bluetooth setelah bluetooth berjalan maka aplikasi akan dapat dijalankan. Kontroller dengan module bluetooth siap untuk mengirim data dari *Odometry* dan *Ultrasonic*.

Data dari sensor ultrasonic berupa jarak antara robot dan benda disekitar robot berupa nilai jarak. Sedangkan data dari *Odometry* berupa nilai jumlah putaran roda dan arah sudut sehingga dari data diatas dapat dijadikan acuan untuk membangun sebuah jalur gerak dari arah robot, jarak yang ditempuh oleh roda(*Odometry*) dan kecepatan dari PWM(Pulse-widh modulation) robot.



Gambar 3.2 Diagram Blok Sistem

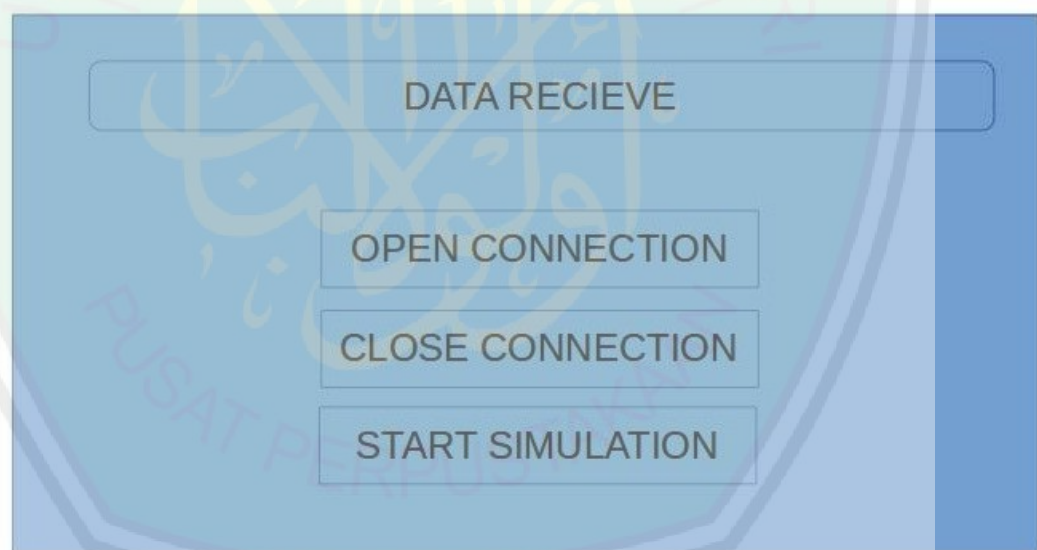
3.1.2 Perancangan Sistem

3.1.1.a Mockup

Dalam membangun Aplikasi generate map berikut ini adalah mockup design tampilan aplikasi pada perangkat android

1. Tampilan Depan

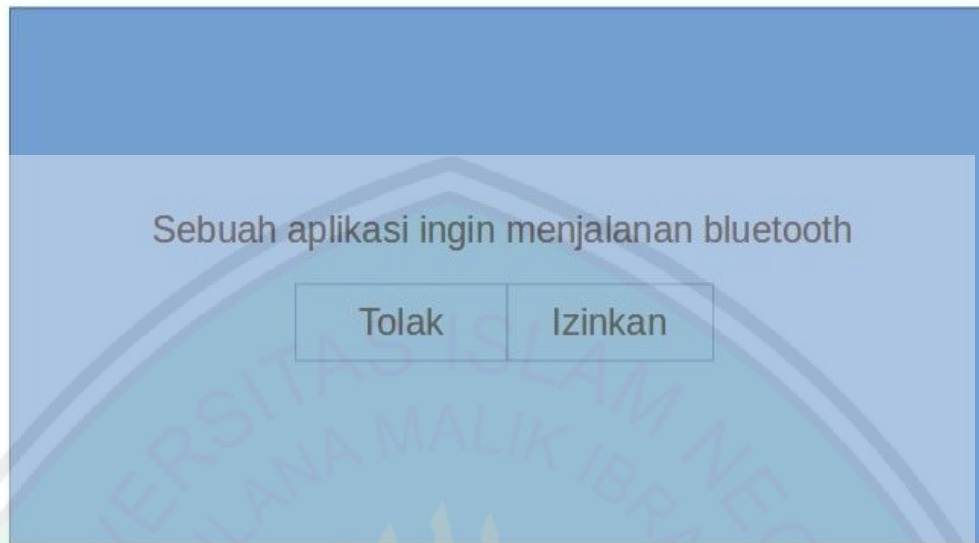
Layout data receive menampilkan nilai input yang dikirim dari mikontroler arduino.



Gambar 3.3 Mockup Tampilan Depan

2. Tampilan perizinan mengaktifkan bluetooth

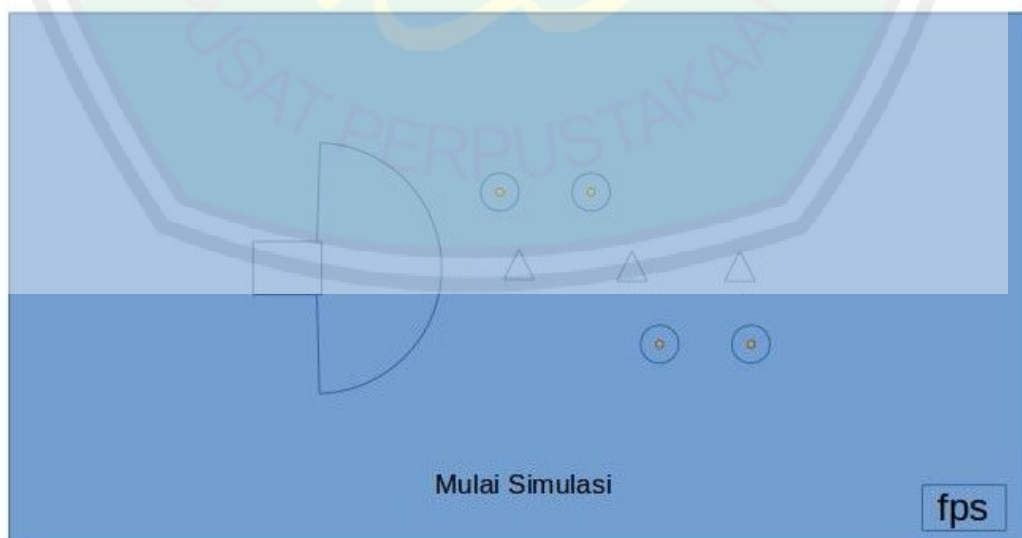
Layout perizinan menyalakan bluetooth untuk *pairing* antara perangkat android dan mikrokontroler arduino



Gambar 3.4 Mockup Tampilan Bluetooth

3. Tampilan Simulasi Jalur Gerak Robot

Layout hasil simulasi setelah memproses data input menggunakan metode EKF-SLAM.



Gambar 3.5 Mockup Tampilan Bluetooth

3.1.2.b Input

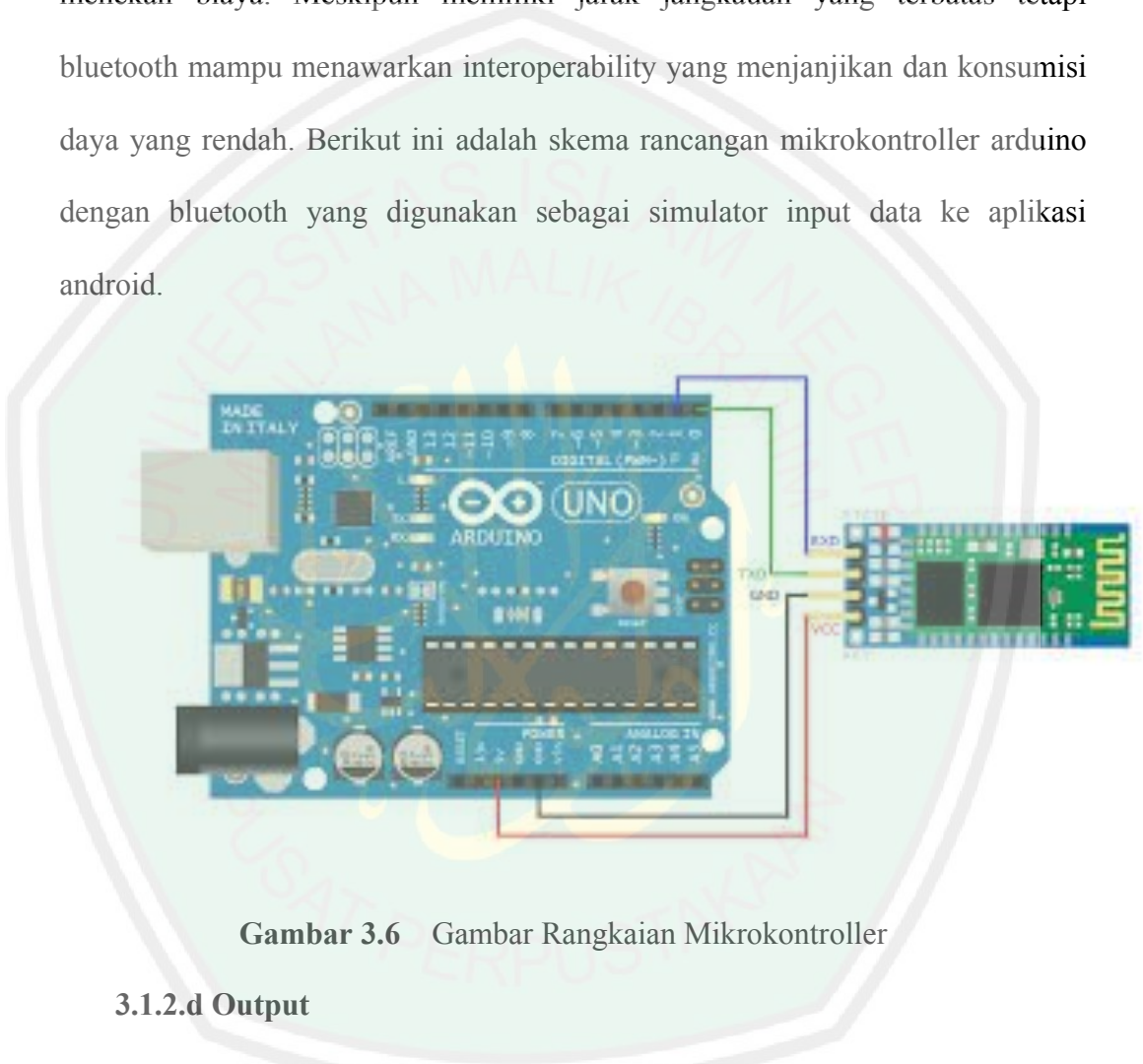
Berdasarkan media yang digunakan, berikut ini penjelasan input sistem:

- a. Switch Button, standart input yang di gunakan oleh user menggunakan perangkat android.
- b. Data sensor Ultrasonic, adalah data yang diberikan melalui sensor yang membaca data halangan dan menghasilkan nilai berupa jarak antara halangan dengan posisi robot.
- c. Odometry, adalah data yang diperoleh dari *rotary encoder* melalui servo robot yang menghasilkan nilai berupa jarak yang ditempuh oleh robot dan juga *steering angle* dari robot.
- d. Data Landmark, adalah benda yang mudah dikenali dan dibedakan dari sebuah lingkungan yang diamati.
- e. Data PWM(Pulse-width modulation), adalah input dari kecepatan robot

3.1.2.c Media Transmisi

Sistem kendali ini menggunakan media transmisi wireless untuk menghubungkan smartphone Android dan sistem mikrokontroler. Ada beberapa pilihan yang dapat digunakan untuk media transmisi wireless seperti infrared, bluetooth, Wi-Fi, dan lain sebagainya tergantung jarak yang diperlukan. Untuk penggunaan sistem pada jarak tidak terlalu jauh, dengan pertimbangan: sistem mikrokontroler diletakan pada bagian tengah rumah, jarak maksimal user dengan sistem mikrokontroler adalah 12 meter.

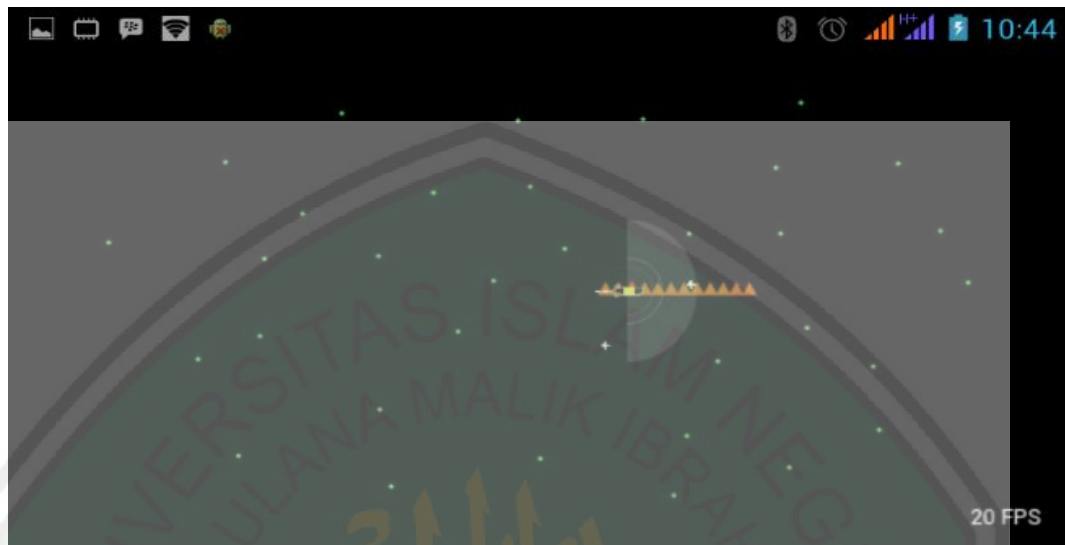
Selain efisiensi jangkauan, biaya menggunakan bluetooth juga jauh lebih murah dari pada Wi-Fi. Menggunakan komunikasi wireless bluetooth lebih menekan biaya. Meskipun memiliki jarak jangkauan yang terbatas tetapi bluetooth mampu menawarkan interoperability yang menjanjikan dan konsumsi daya yang rendah. Berikut ini adalah skema rancangan mikrokontroler arduino dengan bluetooth yang digunakan sebagai simulator input data ke aplikasi android.



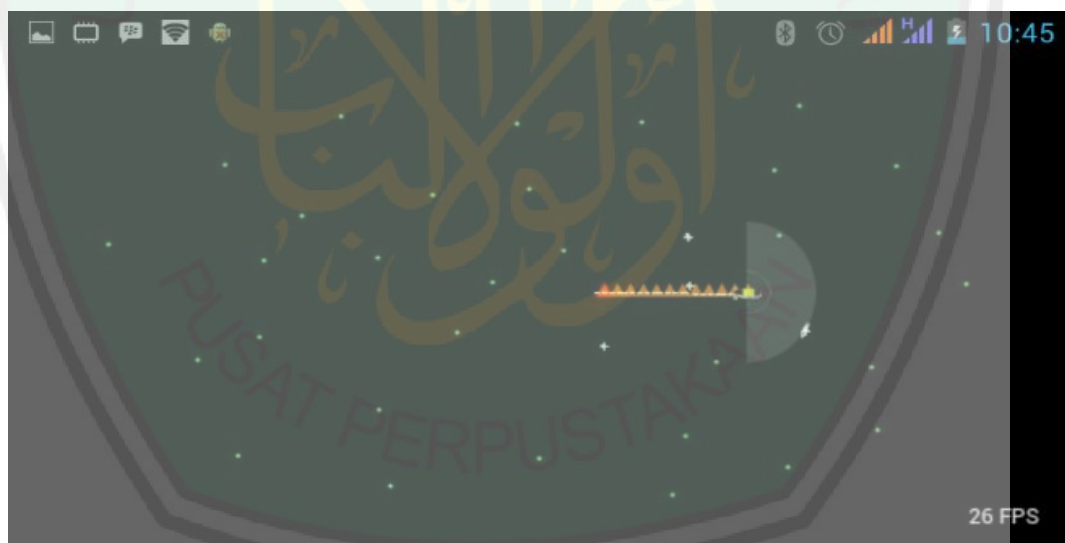
Gambar 3.6 Gambar Rangkaian Mikrokontroler

3.1.2.d Output

Vektor adalah hasil input dari data mikrokontroler sehingga perlu diubah kembali untuk dapat dibaca pada aplikasi di android dalam bentuk koordinat. Output utama yang diberikan oleh aplikasi sistem *generate map* ini adalah sebuah tampilan berupa jalur gerak robot dan halangan pada perangkat android.



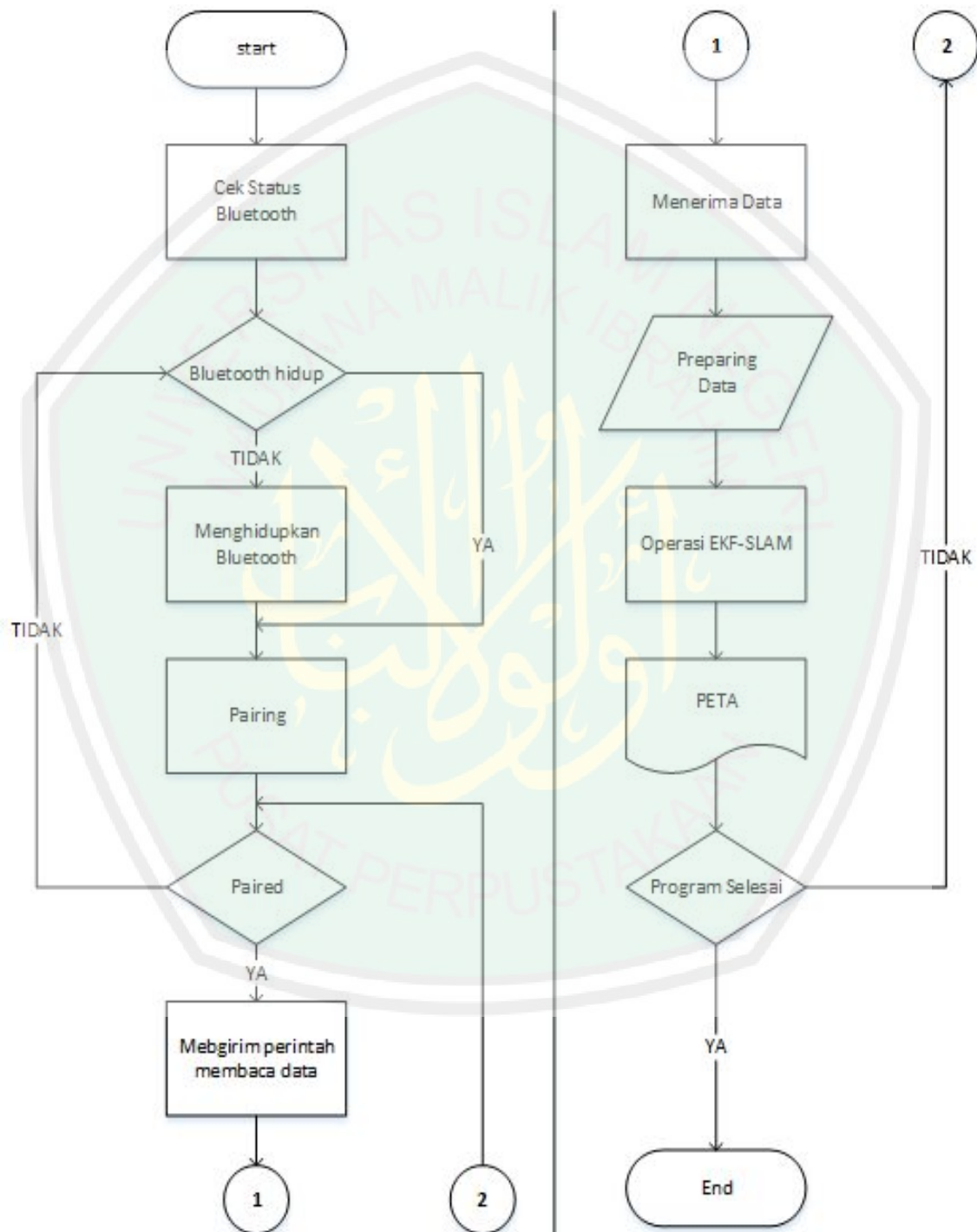
Gambar 3.7 Output Aplikasi 1



Gambar 3.8 Output Aplikasi 2

Tampilan output yaitu jalur yang menghasilkan bentuk halangan dan lokasi dari robot, kemudian juga menampilkan rute yang dilalui oleh robot tersebut.

3.1.3 Flowchart Aplikasi Mobile

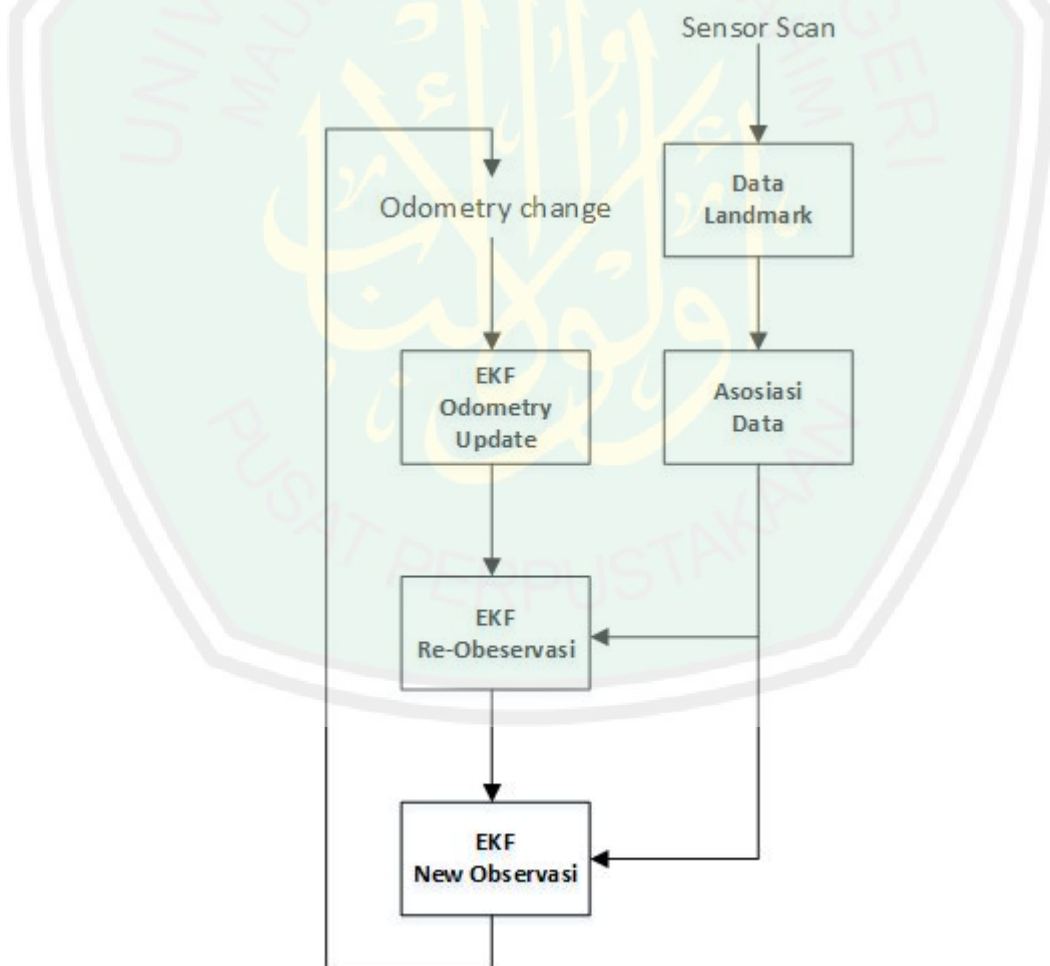


Gambar 3.9 Flowchart Aplikasi pada perangkat Android

3.2 Proses EKF-SLAM

Setelah melihat alur dari mobile apps, proses setelah aplikasi menerima data adalah mengolah data tersebut. Dalam algoritma ekf-slam terdapat beberapa step yang harus dilalui sebelum menghasilkan output di aplikasi android.

Berikut ini adalah alur proses dari metode EKF-SLAM, dimulai dari bagian sensor scan yang mengirimkan data landmark dan posisi robot hingga proses re-observasi dan update.



Gambar 3.10 Proses EKF-Slam

3.3 Algoritma *Extended Kalman Filter (EKF-SLAM)*

Penerapan manual algoritma *Extended Kalman Filter* untuk membangun sebuah rute gerak robot dimulai dari menentukan posisi pergerakan robot

$$\mathbf{x}_{v_k} = \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix}$$

Misal data dari mikrokontroler akan dikirim setiap 5 detik. Kecepatan konstan robot 1m/s dan panjang antara roda depan dan belakang robot adalah 30cm. Maka dapat diambil nilai B adalah 0.3m, V_k adalah 1m/s dan nilai ΔT adalah 5s dan akan selalu konstan karena data akan dikirim setiap 5 detik. Data dari sensor robot masih dalam bentuk koordinat kutub, memiliki arah dan sudut. Data tersebut perlu untuk dirubah menjadi koordinat kartesius untuk dimasukkan dalam formula metode EKF-SLAM.

```

Start
  Setup pos_x, pos_y, pos_phi;
  inisialisasi V, dt, G;
  inisialisasi WHEELBASE
  Hitung posisi baru;
    posisi x:pos_x += V*dt*cos(G + pos_phi)
    posisi y:pos_y += V*dt*sin(G + pos_phi)
    pos_phi:
      pi_to_pi(pos_phi + V*dt*sin(G)/WHEELBASE)
  Simpan posisi baru
  Render
Finish;

```

Gambar 3.11 Pseudocode Algoritma EKF-SLAM

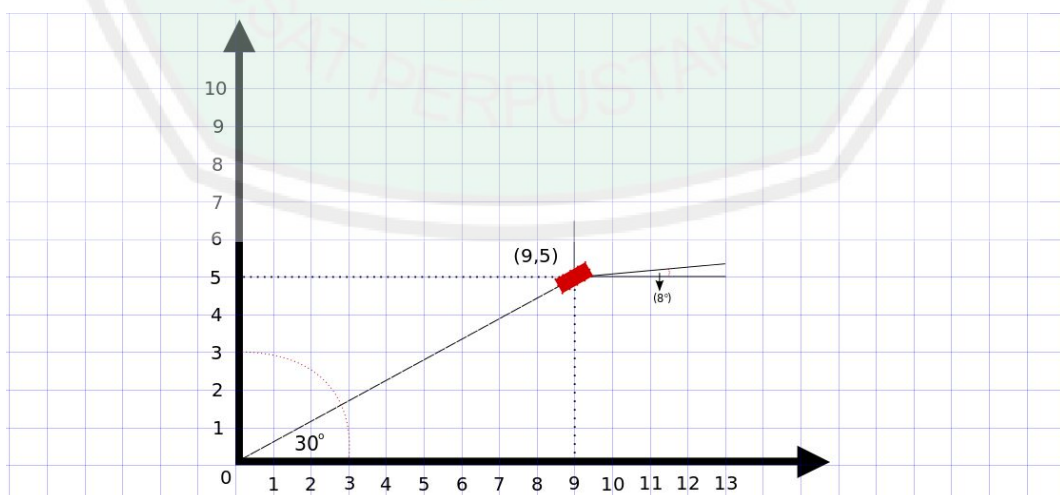
A. Pergerakan 1. Data yang diterima adalah (5m, 30°)

Data pertama dari mikrokontroller dengan nilai odometry 5m dan sudut kendali 30°. Dapat dihitung koordinat kartesiusnya adalah;

$$\begin{aligned} x &= 5 \cdot \cos(30^\circ) & y &= 5 \cdot \sin(30^\circ) \\ &= 4.3 & &= 2.5 \end{aligned}$$

Menghitung pergerakan dengan EKF-Slam;

$$\begin{aligned} X_{vk} &= f_v(X_{vk-1}, U_k) \\ &= [4.3 + 1\text{m/s} \times 5\text{s} \cos(0 + 30^\circ), 2.5 + 1\text{m/s} \times 5\text{s} \sin(0 + 30^\circ), \\ &\quad 0 + 1\text{m/s} \times 5\text{s}/0.3 \sin(30^\circ)] \\ &= [4.3 + 5 \times 0.9, 2.5 + 5 \times 0.5, 0 + 5/0.3 + 0.5] \\ &= [9, 5, 8^\circ] \end{aligned}$$



Gambar 3.11 Manual EKF-SLAM pergerakan pertama

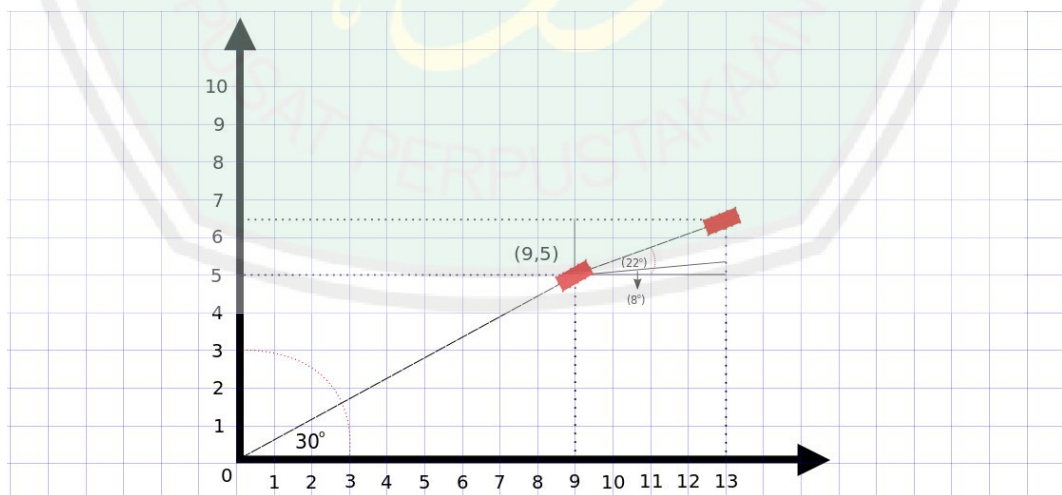
B. Pergerakan 2. Data yang diterima adalah (5m, 22°)

Data kedua dari mikrokontroller dengan nilai odometry 5m dan sudut kendali 22°. Dapat dihitung koordinat kartesiusnya adalah;

$$\begin{aligned} x &= 5 \cdot \cos(22^\circ) & y &= 5 \cdot \sin(22^\circ) \\ &= 9 & &= 2.5 \end{aligned}$$

Menghitung pergerakan dengan EKF-Slam;

$$\begin{aligned} X_{V_k} &= f_v(X_{V_{k-1}}, U_k) \\ &= [9 + 1\text{m/s} \times 5\text{s} \cos(8^\circ + 22^\circ), 4 + 1\text{m/s} \times 5\text{s} \sin(8^\circ + 22^\circ), \\ &\quad 8^\circ + 1\text{m/s} \times 5\text{s}/0.3 \sin(22^\circ)] \\ &= [9 + 5 \times 0.9, 4 + 5 \times 0.4, 8 + 5/0.3 + 0.4] \\ &= [13, 6.5, 14^\circ] \end{aligned}$$



Gambar 3.12 Manual EKF-SLAM pergerakan kedua

3.4 Kebutuhan Sistem

Kebutuhan sistem adalah kebutuhan minimum yang dibutuhkan untuk mengembangkan aplikasi dan juga untuk melakukan uji coba pada aplikasi agar berjalan dengan baik dari segi hardware maupun software.

3.4.1 Kebutuhan Perangkat Keras (*Hardware*)

Dalam membangun aplikasi ini, kebutuhan minimum hardware yang diperlukan adalah, Intel Pentium Dual Core processor (2.1 GHz, 800 MHz FSB, 1 MB L2 cache), 1GB DDR2, 160GB HDD, Wifi Acer Nply 802.11b/g/n.

Sedangkan perangkat keras yang digunakan sebagai simulasi robot adalah

1. Arduino UNO R3 sebagai mikrocontroller



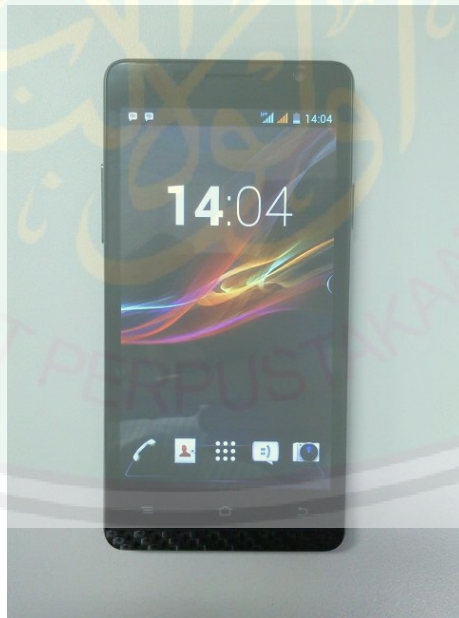
Gambar 3.13 Arduino UNO Board

2. dan DFBluetooth R3 sebagai module wireless untuk mengirimkan data.



Gambar 3.14 DF Robot Bluetooth V3

3. Perangkat Android yang digunakan adalah android minimal versi 2.3 dan yang digunakan untuk penelitian ini adalah Evercross A26C versi 4.2 Jelly Bean.



Gambar 3.15 Evercross A26C

3.4.2 Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan untuk mengembangkan aplikasi;

1. JDK, *Java Development Kit* adalah program *development environment* untuk menulis *Java applets* dan aplikasi.
2. ADT, *Android Development Tools* adalah plugin Eclipse IDE untuk membangun aplikasi Android.
3. MDS Applaud phonegap, adalah plugin eclipse yang digunakan untuk membangun aplikasi hybrid dengan menggunakan library phonegap
4. Arduino IDE, adalah IDE yang digunakan untuk mengembangkan aplikasi berbasis arduino board.

Kebutuhan Software untuk mengembangkan aplikasi lainnya adalah OS Ubuntu 13.10

Saucy Salamander, Kingsoft Writer, Inkscape.

3.4.3. Kebutuhan *Device* Minimum

Berikut ini merupakan daftar spesifikasi *device* yang harus dipenuhi untuk menjalankan aplikasi ini.

Tabel 3.2 Kebutuhan *Device*

Kebutuhan	Spesifikasi Minimum	Spesifikasi Rekomendasi
Versi Android	2.2	2.3 keatas
GPU	PowerVR SGX 100	PowerVR SGX 530
Memori	1 GB	2GB

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada bab ini membahas tentang implementasi dari perencanaan yang telah dibuat. Serta melakukan pengujian terhadap *aplikasi generate map* untuk mengetahui apakah *aplikasi* tersebut dapat berjalan baik sesuai dengan yang diharapkan pada perangkat mobile khususnya android.

4.2 Implementasi Algoritma *EKF-SLAM* pada *Aplikasi*

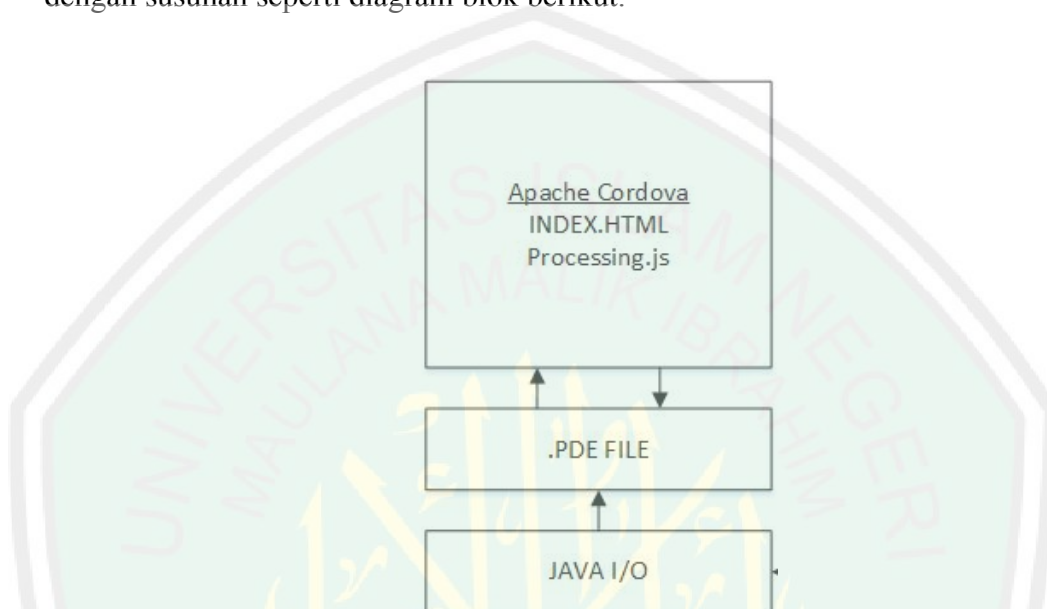
Implementasi metode ekf-slam diterapkan pada aplikasi android. Dalam menentukan posisi dan arah hadap robot menggunakan EKF-SLAM menggunakan algoritma sebagai berikut

```

Start
  Setup pos_x, pos_y, pos_phi;
  inisialisasi V, dt, G;
  inisialisasi WHEELBASE
  Hitung posisi baru;
  posisi x:
    - pos_x += V*dt*cos(G + pos_phi)
  posisi y:
    - pos_y += V*dt*sin(G + pos_phi)
  pos_phi:
    - pi_to_pi(pos_phi + V*dt*sin(G)/WHEELBASE)
  Simpan posisi baru
  Render
Finish;

```

Gambaran secara garis besar Metode EKF-Slam dijalankan menggunakan plugin phonegap dan dalam bentuk ekstensi *.pde* yaitu ekstensi untuk processing.js dengan susunan seperti diagram blok berikut.



Gambar 4.1 Diagram Blok Aplikasi Writer I/O

4.2.1 Program Processing ekfslam.pde

Algoritma *EKF-SLAM* pada file *ekfslam.pde* dalam *aplikasi* ini akan diimplementasikan secara umum dengan *fungsi* seperti berikut :

```

void update() {
  /*
  Menghitung posisi yang benar dari robot dan posisi estimasinya
  */
  // Menghitung kendali gerak
  compute_steering();

  // Melakukan perulangan: jika titik final telah dicapai, kembali ke titik awal
  if (iwp == -1) {
    iwp = 0;
    if (iloop > 1) {

```

```
        iloop -= 1;
    } else {
        noLoop();
    }
}

// Membuat model dari robot atau vehicle
vehicle_model();

// Tahap untuk prediksi pada EKF
predict();

dtsum += dt;
if (dtsum < DT_OBSERVE)
    return;
dtsum = 0.0;

// Observasi pada data landmark yang ada
get_observations();

// Melihat data table untuk di update ke feature selanjutnya
data_associate_known();

// update EKF
batch_update();

// menambahkan kondisi baru dari hasil observasi
augment();

// menambahkan estimasi jalur yang telah dilalui oleh roboto
store_data();
}
```

Berikut akan dijelaskan penggunaan *method* pada tabel 4.1

Tabel 4.1 Tabel Keterangan *Method* Class pada Aplikasi processing

No	Method	Keterangan
1.	<code>Compute_steering()</code>	Bertugas untuk menghitung kontrol gerak robot berdasarkan index <i>waypoint</i> dan kemudian mengarahkan ke titik selanjutnya
2.	<code>Vechile_model()</code>	Menghitung posisi dan kecepatan robot yang tetap dan penerapan model EKF-Slam simulasi
3.	<code>Add_random_noise()</code>	Menambahkan noise untuk observasi simulasi
4.	<code>Predict()</code>	Fungsi untuk memprediksi posisi dan keadaan robot dalam simulasi
5.	<code>Get_observation()</code>	Mengobservasi landmark yang sudah di set sebelumnya
6.	<code>Add_observation_noise()</code>	Menambahkan random noise untuk fungsi observasi
7.	<code>Data_association_know()</code>	Mengatur tabel fitur/benda untuk proses observasi dengan data yang sudah ada
8.	<code>Batch_update()</code>	Memberikan fitur index atau urutan

		dan <i>range-bearing</i> (jarak) benda terhadap robot.
9.	Augment ()	Menambahkan data posisi dan keadaan robot bila observasi baru
10.	Store_data ()	Menambahkan estimasi jalur yang dilalui oleh robot.

4.2.2 Program Map.html

Pada halaman ini, tag html canvas berfungsi untuk memanggil file berekstensi .pde yaitu file yang digunakan untuk memproses metode EKF-SLAM. Untuk menjalankannya dibutuhkan library processing.js

```
<!DOCTYPE HTML>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <meta charset="utf-8">
</head>
<style>
.btn-on {
  z-index:1;
  margin-bottom:20px;
}
</style>
<body style="background:#000;">
  <div id="content">
    <script src="processing-1.4.1.js"></script>
    <canvas id="__processing0"
data-processing-sources="ekfslam.pde matrix.pde"> </canvas>
  </div>
</body>
```

```
</html>
```

4.2.3 Android Phonegap Injeksi data Java I/O

Program Java I/O berfungsi untuk mengupdate data yang dibutuhkan oleh aplikasi. Data yang dimaksud adalah data fitur-fitur dan data *path* koordinat dari robot. Variabel-variabel lainya seperti kecepatan, nilai wheelbase, kontrol interval sudah terinisialisasi pada file ekfslam.pde untuk menjalankan program processing. Selain itu program ini juga mengatur koneksi bluetooth dengan robot untuk proses *pairing* dan dan transfer data.

Tabel 4.2 Tabel Keterangan *Method* Class pada aplikasi Android Bluetooth dan Start

No	Method	Keterangan
1.	<code>onCreate()</code>	Menampilkan tombol UI mulai untuk menjalanan aplikasi ketika data bluetooth sudah diterima.
2.	<code>openButton</code>	Digunakan untuk menghubungkan perangkat android dengan mikrokontroller arduino
3.	<code>sendButton</code>	Berfungsi untuk mengirim data ke mikrokontroller arduino. Digunakan nanti bila perangkat android memiliki fungsi untuk mengontrol mikrokontroller
4	<code>closeButton</code>	Digunakan untuk menutup koneksi

		antara mikrokontroller arduino dan bluetooth.
5	<code>BtnMulai</code>	Digunakan untuk memulai menampilkan output simulasi setelah data dari mikrokontroller diterima.
6	<code>findBT()</code>	Digunakan untuk mencari <i>Bluetooth Device</i> dari robot
7	<code>openBT()</code>	Berfungsi untuk membuka koneksi bluetooth dari perangkat android
8	<code>beginListenForData()</code>	Berfungsi untuk mengecek apa ada data yang diterima oleh perangkat android
9	<code>sendData()</code>	Fungsi untuk mengirim data dari android ke mikrokontroller arduino
10	<code>startMap()</code>	Menjalankan output simulasi berdasarkan data yang telah diterima
11	<code>fixJellyBeanIssues()</code>	Fungsi untuk memperbolehkan akses data eksternal. Berlaku untuk OS Android versi JellyBean.

Pada Class PrograssTask ini berfungsi untuk implementasi I/O yang bertugas untuk meng-*inject* data melalui bluetooth yang berjalan menggunakan Thread.

```

public class PrograssTask extends AsyncTask<Integer,
Integer, Void> {
    @Override
    protected Void doInBackground(Integer... params) {
        // TODO Auto-generated method stub
        Writing();
        return null;
    }
    private void Writing() {
        // TODO Auto-generated method stub
        try {
            String[] lines = new String[888];
            File internalStorage =
Environment.getDataDirectory();
            File dir = new File (internalStorage + "/myPde");

            File file = new File(dir, "ekfslam.pde");
            File file_temp = new File(dir,
"ekfslam_temp.txt");

            FileWriter fw = new FileWriter(file_temp, true);
            BufferedWriter bw = new BufferedWriter(fw);
            LineNumberReader lnr = new LineNumberReader(new
FileReader(file));

            file_temp.createNewFile();

            int i = 0;
            String Line = "";
            while (true) {
                Line = lnr.readLine();
                if (Line != null) {
                    lines[i]=Line;
                    i++;
                } else {
                    break;
                }
            }
            for (int j = 0; j < 50; j+=5) {
                lines[95] = lines[95]+"\n\t\t Data "+j;
            }
        }
    }
}

```

```

    if (file_temp.exists()) {
    } else {
        file_temp.createNewFile();
    }

    for (i = 0; i < lines.length; i++){
        bw.write(lines[i]);
        bw.newLine();
        System.out.println(lines[i]);
    }
    file.delete();
    file_temp.renameTo(file);
    bw.close();
    lnr.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

Tabel 4.3 Tabel Keterangan *Method* Class pada Aplikasi Write java I/O android

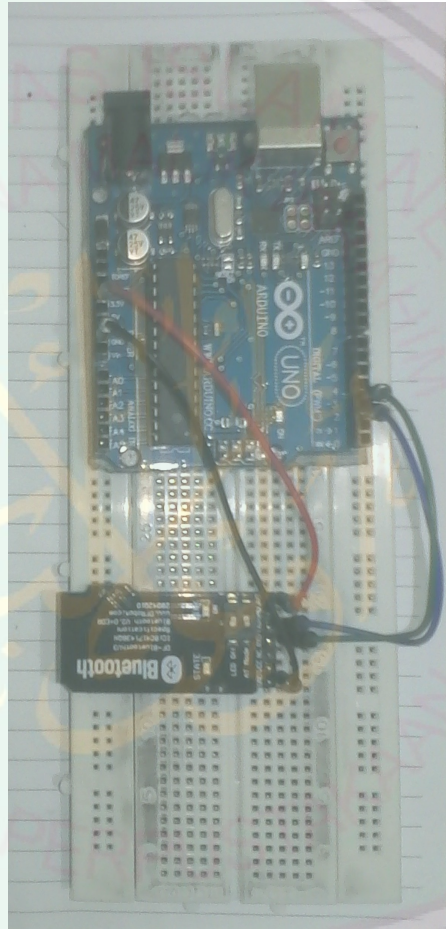
No	Method	Keterangan
1.	doInBackground()	Menjalankan class writing dalam proses background untuk mengupdate file pde dengan diinject menggunakan java I/O.
2	Writing()	Membaca semua string per baris pada file pde dan menyimpan dalam bentuk array. Lalu merubah nilai array pada index yang ditentukan. Kemudian menulis ulang array tersebut

Program tersebut membaca jumlah baris pada file .pde. Kemudian menentukan baris mana yang akan dirubah. Dalam kasus tersebut baris yang dirubah adalah baris dimana nilai variabel waypoint(koordinat jalur) berada. Dengan begitu data yang masuk dari bluetooth dapat langsung di *inject* kedalam file .pde dan siap dibaca oleh map.html



4.2.4 Sistem Mikrokontroler Simulator *Vechile* untuk Mengirim Data

Sedangkan bentuk simulasi *vechile* sebagai pengirim data menggunakan mikrokontroler arduino dengan module bluetooth sebagai pengirim melalui wireless ke aplikasi yang men-*generate* jalur gerak robot pada perangkat android.



Gambar 4.2 Mikrokontroler Arduino dan Bluetooth

4.2.5 Program untuk Mengirim Data di Arduino Uno R3

```
#include <SoftwareSerial.h>
int bluetoothTx = 2;
int bluetoothRx = 3;
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);
void setup()
{
  //Setup usb serial connection to computer
  Serial.begin(9600);

  //Setup Bluetooth serial connection to android
  bluetooth.begin(115200);
  bluetooth.print("$$$");
  delay(100);
  bluetooth.println("U,9600,N");
  bluetooth.begin(9600);
}
void loop()
{
  //Read from bluetooth and write to usb serial
  if(bluetooth.available())
  {
    char toSend = (char)bluetooth.read();
    Serial.print(toSend);
  }

  //Read from usb serial to bluetooth
  if(Serial.available())
  {
    char toSend = (char)Serial.read();
    bluetooth.print(toSend);
  }
}
```


Tabel 4.4 Tabel Keterangan *Method Class* pada Aplikasi Arduino

No	Method	Keterangan
1.	setup()	Inisialisasi variable usb serial koneksi ke komputer dan bluetooth serial untuk android
2	loop()	Berisi fungsi membaca data dari bluetooth dan menulis data ke USB serial. Membaca dari USB serial ke bluetooth

4.3 Hasil pengujian Algoritma *EKF-SLAM*

Berikut akan dijelaskan tentang hasil pengujian dari algoritma *EKF-SLAM* dalam bentuk tabel pada tabel dibawah ini. Tabel “Menghitung pergerakan dengan *EKF-SLAM* sesuai dengan alur pada penghitungan manual metode dimana alur kerja program sesuai dengan pembahasan sebelumnya.

Data dibawah ini merupakan simulasi data yang dikirim dengan parameter masukan kecepatan robot 1m/s, WHEELBASE 0.3m, dan simulasi pergerakan robot 5m/s. Maka koordinat yang dihasilkan bila menerima 7 masukan data adalah sebagai berikut.

Data Masukan	Pergerakan Robot (vektor)	$x.\cos(30^\circ)$	$y.\sin(30^\circ)$
Masukan 1	5m, 30°	4.3	2.5
Menghitung pergerakan dengan <i>EKF-Slam</i>			
X_{vk}	$f_v(X_{vk-1}, U_k)$	[9, 5, 8°]	
Data Masukan	Pergerakan Robot (vektor)	$x.\cos(22^\circ)$	$y.\sin(22^\circ)$
Masukan 2	5m, 22°	9	2.5
Menghitung pergerakan dengan <i>EKF-Slam</i>			

X_{vk}	$f_v(X_{vk-1}, U_k)$	[13, 6.5, 14°]
----------	----------------------	-----------------

Data Masukan	Pergerakan Robot (vektor)	$x.\cos(22^\circ)$	$y.\sin(22^\circ)$
Masukan 3	5m, 45°	3.5	3.5
Menghitung pergerakan dengan EKF-Slam			
X_{vk}	$f_v(X_{vk-1}, U_k)$	[20, 14.2, 31°]	

Data Masukan	Pergerakan Robot (vektor)	$x.\cos(30^\circ)$	$y.\sin(30^\circ)$
Masukan 4	5m, 30°	4.3	2.5
Menghitung pergerakan dengan EKF-Slam			
X_{vk}	$f_v(X_{vk-1}, U_k)$	[29, 11.5, 32°]	

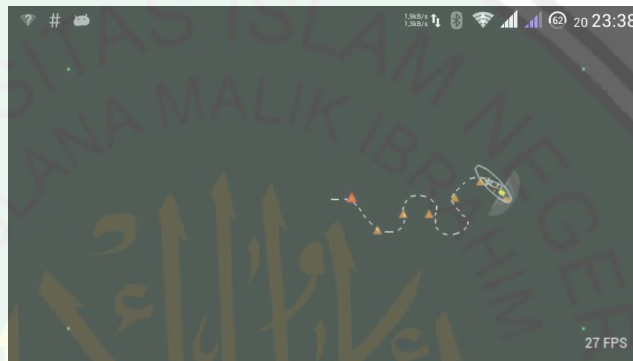
Data Masukan	Pergerakan Robot (vektor)	$x.\cos(30^\circ)$	$y.\sin(30^\circ)$
Masukan 5	5m, 30°	4.3	2.5
Menghitung pergerakan dengan EKF-Slam			
X_{vk}	$f_v(X_{vk-1}, U_k)$	[38, 16.5, 33.2°]	

Data Masukan	Pergerakan Robot (vektor)	$x.\cos(60^\circ)$	$y.\sin(60^\circ)$
Masukan 6	5m, 60°	2.5	4.3
Menghitung pergerakan dengan EKF-Slam			
X_{vk}	$f_v(X_{vk-1}, U_k)$	[40.2, 25.8, 34.7°]	

Data Masukan	Pergerakan Robot (vektor)	$x.\cos(22^\circ)$	$y.\sin(22^\circ)$
Masukan 7	5m, 22°	9	2.5
Menghitung pergerakan dengan EKF-Slam			
X_{vk}	$f_v(X_{vk-1}, U_k)$	[44.5, 8, 35.7°]	

Uji coba langsung pada aplikasi dengan menggunakan beberapa parameter kontrol yang berbeda menghasilkan *path* atau jalur gerak robot yang berbeda. Hasil dari ujicoba tersebut dapat dilihat perbedaan output signifikan dari jalur gerak robot

1. Uji coba 1



Dengan parameter:

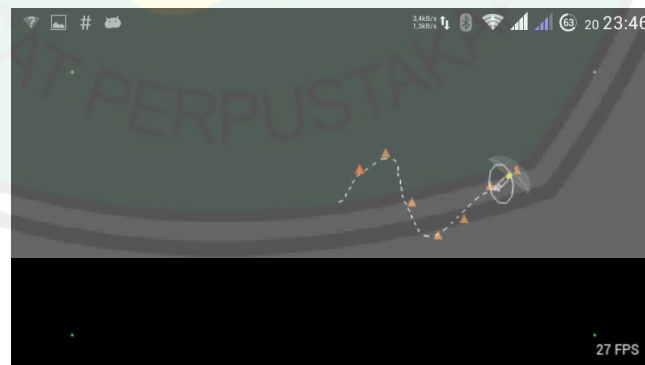
Coordinates : {1,2},{2,3},{3,0},{4,-2},{5,-1},{6,1},{7,2}

FWD_SPEED = 1;

MAXG = $45 \cdot \text{PI} / 180.0$;

WHEELBASE = 0.5;

2. Uji coba 2



Dengan parameter:

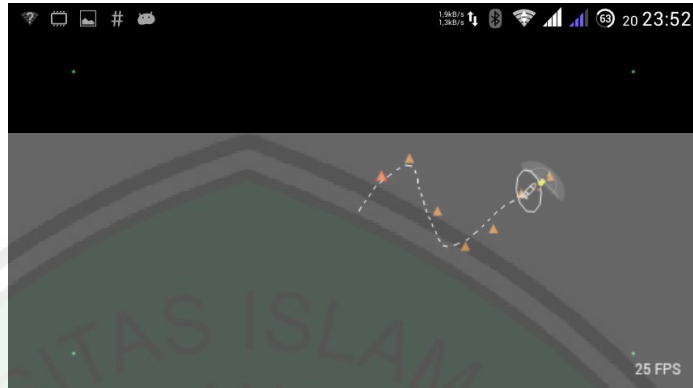
Coordinates : {1,2},{2,3},{3,0},{4,-2},{5,-1},{6,1},{7,2}

FWD_SPEED = 1;

MAXG = $90 \cdot \text{PI} / 180.0$;

WHEELBASE = 0.5;

3. Uji coba 3



Dengan parameter:

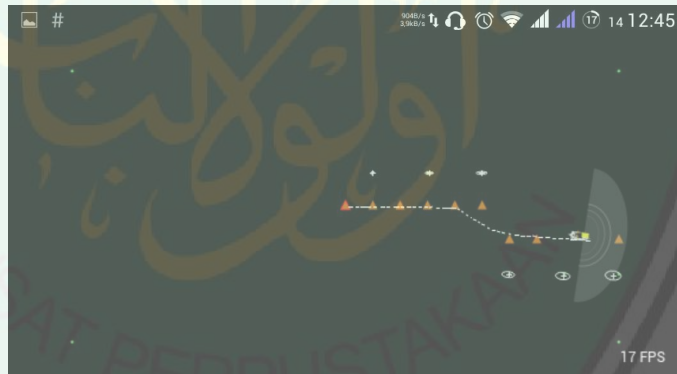
Coordinates : $\{1,2\},\{2,3\},\{3,0\},\{4,-2\},\{5,-1\},\{6,1\},\{7,2\}$

FWD_SPEED = 1;

MAXG = $180 \cdot \text{PI} / 180.0$;

WHEELBASE = 0.5;

4. Uji coba 4



Dengan parameter:

Coordinates : $\{1,0\},\{2,0\},\{3,0\},\{4,0\},\{5,0\},\{6,-1\},\{7,-1\},\{9,-1\}$


FWD_SPEED = 1;

MAXG = $180 \cdot \text{PI} / 180.0$;

WHEELBASE = 0.5;

Landmark: $\{1,2\},\{3,2\},\{5,2\},\{6,-2\},\{8,-2\},\{9,-2\}$,

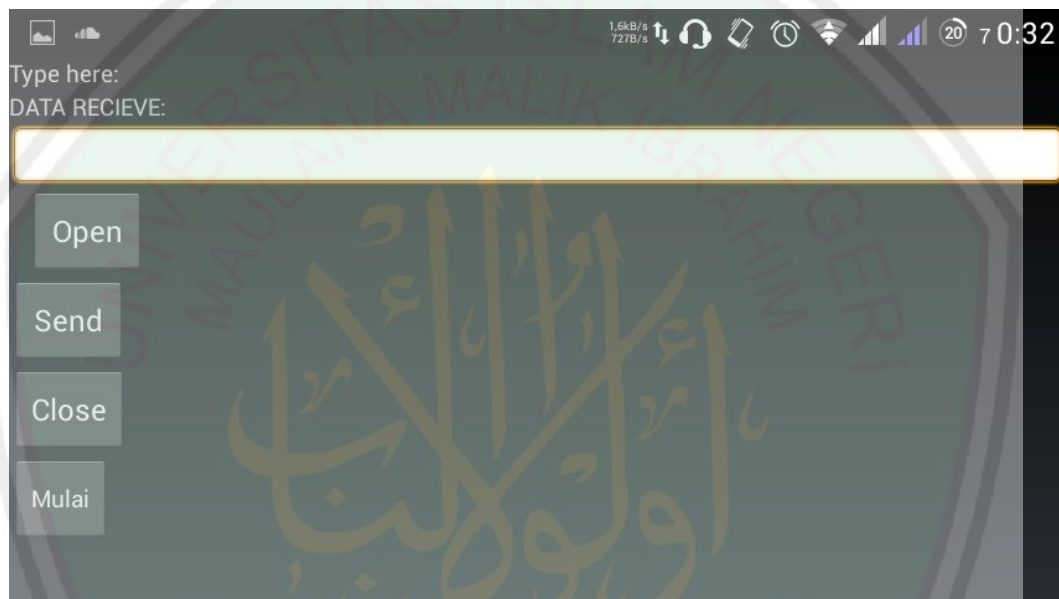
Tabel 4.5 Tabel Hasil Pengujian Algoritma *EKF-SLAM*

Aplikasi	Simulasi	Keterangan
Algoritma	Extended Kalman Filter	
Parameter	FWD_SPEED = 6;	Vechile forward speed(m/s)
	MAXG = $90.0 \cdot \text{PI} / 180.0$	Maximum Steering angle
	WHEELBASE = 4.0;	Vechile wheelbase (m)
	DT_CONTROLS = 0.025;	Time interval antara controll sinyall (s)
Frame/s	25 - 30 FPS	Frame Render
Landmark	{1,1}, {3,1}, {6,-2}, {8,-2}, {9,-2}	
Wapoint	{1,0}, {2,0}, {3,0}, {4,0}, {5,0}, {6,-1}, {7,-1}, {9,-1}	
Hasil		

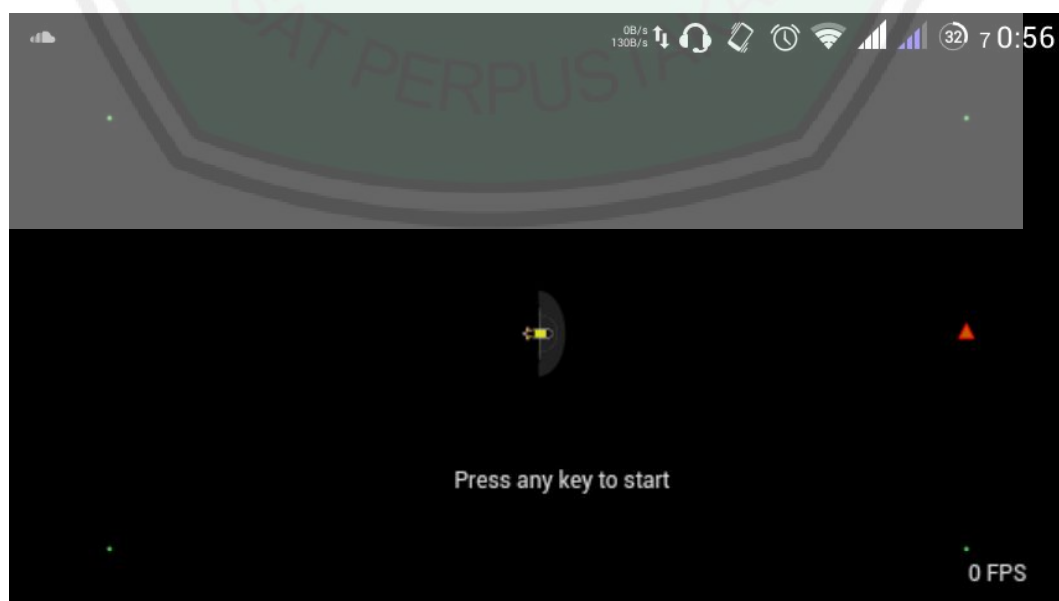
4.4 Implementasi Aplikasi

Berikut adalah tampilan *aplikasi* yang telah selesai dibuat. Tombol open digunakan untuk menghubungkan perangkat android dengan mikrokontroller arduino untuk menerima data yang akan digunakan sebagai parameter simulasi. Tombol Send digunakan untuk mengirim data test ke mikrokontroller arduino.

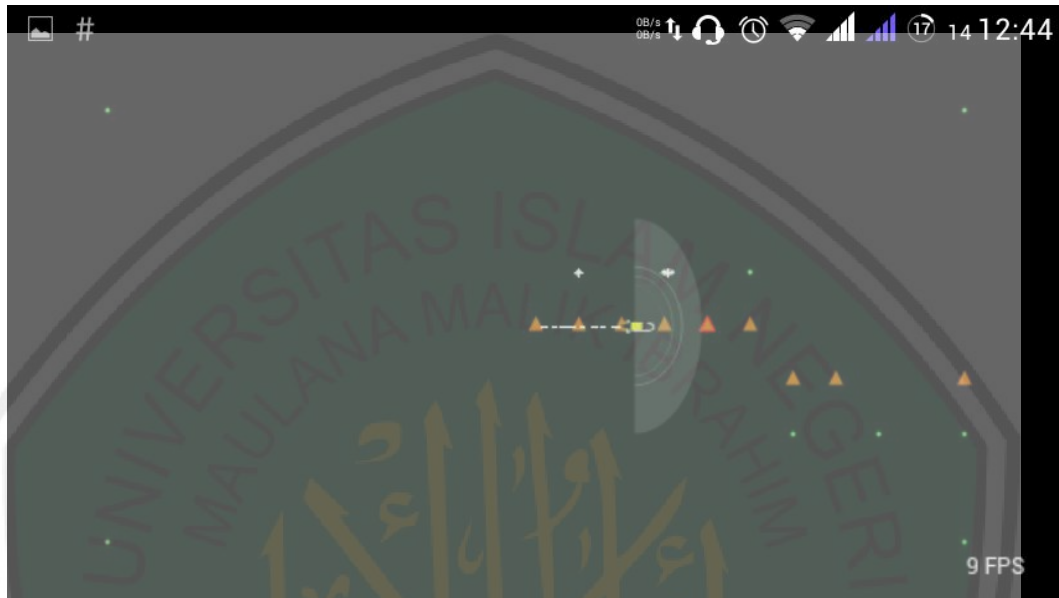
Sedangkan tombol close digunakan untuk memutus koneksi bluetooth antara mikrokontroler dengan perangkat android. Setelah data selesai diterima, selanjutnya dapat memulai menampilkan output berdasarkan data dari mikrokontroler dengan menekan tombol Mulai.



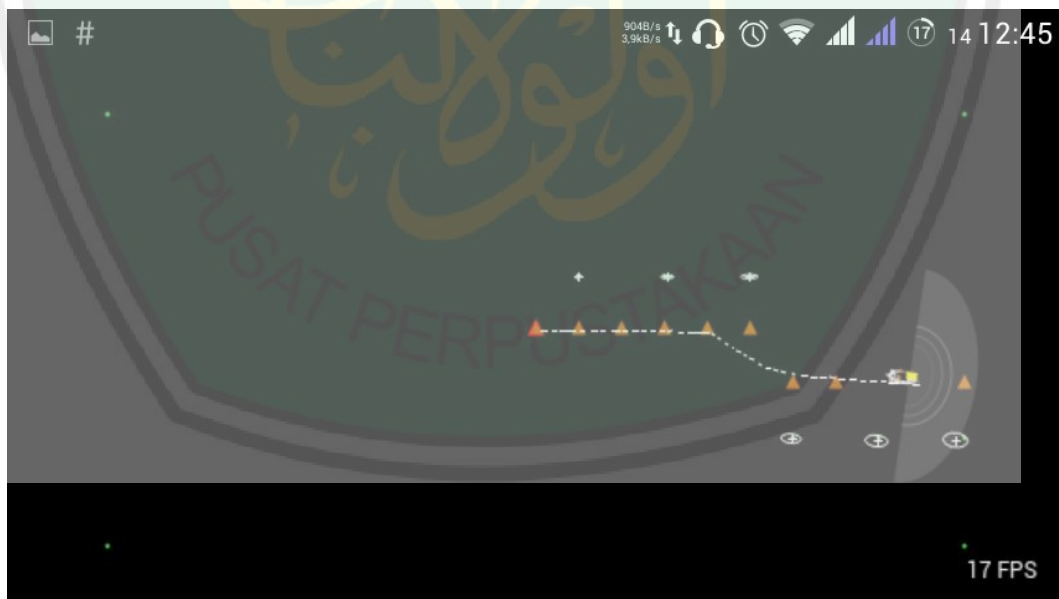
Gambar 4.3 Tampilan depan



Gambar 4.4 Tampilan Awal



Gambar 4.5 Tampilan saat Simulasi Robot



Gambar 4.6 Tampilan Saat Simulasi Selesai

4.5 Pengujian Aplikasi pada Perangkat Android

Berikut akan dijelaskan tentang hasil pengujian dari aplikasi *system generate map* pada *smartphone* dalam bentuk tabel pada tabel 4.4.

Tabel 4.6 Tabel Hasil Pengujian *aplikasi pada perangkat android*

No	Merk	Versi	GPU	CPU	RAM	Keterangan
1.	Evercoss A26C	v4.2 (Jelly Bean)	Adreno 203	Dual Core 1,2 GHz	512 MB RAM	Sistem berjalan baik. Frame rate aplikasi simulasi masih terlalu lambat berkisar 28-32 FPS

4.6 Integrasi Dalam Islam

Pada surah al-Kahf (18) ayat ke 17, berfirman,

﴿ وَتَرَى الشَّمْسَ إِذَا طَلَعَتْ تَزْوُرُ عَنْ كَهْفِهِمْ ذَاتَ الْيَمِينِ وَإِذَا غَرَبَتْ
تَقْرِبُهُمْ ذَاتَ الشِّمَالِ وَهُمْ فِي فَجْوَةٍ مِّنْهُ ذَلِكَ مِنْ آيَاتِ اللَّهِ مَن
يَهْدِ اللَّهُ فَهُوَ الْمُهْتَدِ وَمَن يُضِلِّ فَلَن تَجِدَ لَهُ وَلِيًّا مُّرْشِدًا ﴾
﴿١٧﴾

"Dan engkau akan melihat matahari ketika terbit, cenderung ke kanan dari gua mereka; dan apabila ia terbenam, meninggalkan mereka ke arah kiri, sedang mereka berada dalam satu lapangan gua itu. Yang demikian ialah dari tanda-tanda (yang membuktikan kekuasaan) Allah. Sesiapa yang diberi hidayah petunjuk oleh Allah, maka dia lah yang berjaya mencapai kebahagiaan; dan sesiapa yang disesatkanNya maka engkau tidak sekali-kali akan beroleh sebarang penolong yang dapat menunjukkan (jalan yang benar) kepadanya." Al-Quran adalah petunjuk hanya bagi orang yang takwa.

Setiap manusia harus memiliki panduan dalam hidup agar dapat memahami dan memperoleh pelajaran dari padanya. Sebuah Sistem petunjuk atau sistem navigasi dalam dunia IT sangat dibutuhkan untuk mempermudah manusia menentukan jalan yang tepat. Sistem navigasi membuat perjalanan menjadi tenang karena kita tahu kemana tujuan kita, begitu juga Al-Qur'an, adalah sebagai petunjuk untuk manusia yang bertakwa hidup di dunia.

Berdasarkan informasi diatas mendasari penelitian ini untuk membangun aplikasi sistem generate map untuk daerah yang sulit dijangkau berdasarkan parameter data sensor yang dikirim oleh robot dengan menerapkan metode Extende Kalman Filter Simultaneous localization and Mapping (EKF-SLAM). Sehingga dengan adanya aplikasi ini diharapkan dapat membantu dalam mengamati daerah yang sulit dijangkau ataupun yang belum terjamah oleh manusia dengan bantuan robot. Dengan adanya aplikasi moblie monitoring controll maka dapat dihasilkan system kendali yang lebih cepat dan *real time*, terutama dalam menyikapi dunia teknologi saat ini yang menuntut kualitas, kecepatan dan ketepatan.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan uraian yang telah dijelaskan dalam penelitian ini tentang membangun aplikasi generate map menggunakan metode EKF-SLAM dapat disimpulkan sebagai berikut :

1. Dalam membangun simulasi jalur gerak robot menggunakan algoritma EKF-SLAM pada aplikasi mobile berjalan sesuai dengan yang diharapkan. Ujicoba data 7 masukan dengan parameter yang ditentukan telah menghasilkan output koordinat yang diinginkan.
2. Ujicoba dalam aplikasi menunjukkan perbedaan bentuk jalur gerak robot yang signifikan berdasarkan parameter input yang berbeda. Data parameter ini merupakan bentuk kondali robot yang dapat disesuaikan dengan kebutuhan dan keadaan robot
3. Testing pada aplikasi android secara performa masih belum sempurna. Data yang dikirim ke perangkat android harus disiapkan terlebih dahulu sebelum simulasi dijalankan karena proses menulis data pada file pde thread perlu di close terlebih dahulu. Saat proses close thread tersebut aplikasi android akan mengalami *Crash* pada *UI Thread Phonegap*. Dengan kondisi tersebut data harus diproses satu per satu.

5.2 Saran

Penulis sadar, dalam pembuatan *Aplikasi* ini masih banyak kekurangan yang nantinya perlu untuk dilakukan pengembangan, diantaranya :

1. Mengganti platform phonegap menjadi platform native android menggunakan IDE Processing yang sudah dapat mengcompile program ke dalam perangkat android sehingga pemrosesan dan performa dari aplikasi lebih cepat dan *reliable*.
2. Dalam aplikasi ini hanya menerima data berupa input koordinat dalam membangun jalur gerak robot. Parameter lainnya sudah terinisialisasi pada file pde processing. Untuk lebih menyempurnakan aplikasi diperlukan data koordinat yang real dari sensor sensor yang lebih baik.
3. Simulasi menggunakan phonegap masih belum dapat membangun lingkungan robot secara realtime. Phonegap membutuhkan external file seperti pde processing untuk menjalankan simulasi sehingga saat data ditulis menggunakan java I/O ke dalam file pde thread pada UI phonegap harus dihentikan agar aplikasi tidak terjadi *crash*
4. UI aplikasi ini masih sangat sederhana sehingga akan lebih menarik bila memaksimalkan UI aplikasi ini.

DAFTAR PUSTAKA

- Tim Bailey, Juan Nieto. 2006. *Consistency of the EKF-SLAM Algorithm*. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on
- Blog of Rog 2011. *Simultaneous Localisation and Mapping (HTML5/Processing)*
- Søren Riisgaard and Morten Rufus Blas. 2005. *SLAM for Dummies A Tutorial Approach to Simultaneous Localization and Mapping*
- Megan R. Naminski. 2013. *An Analysis of Simultaneous Localization and Mapping (SLAM) Algorithms*. Macalester College
- Semiloka Perkembangan Robot di Indonesia 2004. "*Peluang dan Tantangan Teknologi Robot di Indonesia*". Ristek.go.id.
- Safaat H, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Abdul Majid bin Aziz Al-Zindani. 1997. *Mukjizat Al-Qur'an dan As-Sunnah tentang IPTEK*. Jakarta Gema Insani Press.

Lampiran 1

1. Kode Program Aplikasi Android

```
package map.skripsi.unite_thread_writer_ui;

public class Unite_TWUActivity extends DroidGap {

    ProgressTask task = new ProgressTask();

    TextView myLabel;

    TextView arrBufferData;

    EditText myTextbox;

    /*Bluetooth Initialize*/

    BluetoothAdapter mBluetoothAdapter;

    BluetoothSocket mmSocket;

    BluetoothDevice mmDevice;

    OutputStream mmOutputStream;

    InputStream mmInputStream;

    Thread workerThread;

    byte[] readBuffer;

    int readBufferPosition;

    int counter;

    volatile boolean stopWorker;

    Button BtnMulai;

    private int retryCount = 0;

    List<String> bufferData = new ArrayList<String>();

    /*String[] bufferData = null;*/

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

    }

}
```

```
Button openButton = (Button)findViewById(R.id.open);
Button sendButton = (Button)findViewById(R.id.send);
Button closeButton = (Button)findViewById(R.id.close);
Button resetButton = (Button)findViewById(R.id.reset);
myLabel = (TextView)findViewById(R.id.label);
arrBufferData = (TextView)findViewById(R.id.arrBufferData);

myTextbox = (EditText)findViewById(R.id.entry);

/*openButton*/
openButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        try {
            findBT();
            openBT();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
});
```

```
/*senButton*/
sendButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        try {
            sendData();
        } catch (IOException e) {
            // TODO Auto-generated catch block

```

```

        e.printStackTrace();
    }
}
});

/*closeButton*/
closeButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        try {
            closeBT();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
});

/*resetButton*/
resetButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        task.resetBT();
    }
});

BtnMulai = (Button) findViewById(R.id.start);
BtnMulai.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String[] dataoke= new String[bufferData.size()];

```

```

        for (int i=0;i<bufferData.size();i++){
            dataoke[i]=bufferData.get(i);
        }
        task.execute(dataoke);
        startMap();
    }
});
}

void findBT() {
    mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    if(mBluetoothAdapter == null){
        myLabel.setText("No bluetooth adapter available");
    }
    if(!mBluetoothAdapter.isEnabled()){
        Intent enableBluetooth = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBluetooth, 0);
    }

    Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
    if(pairedDevices.size()>0){
        for(BluetoothDevice device: pairedDevices){
            if(device.getName().equals("Bluetooth_Bee_V2")){
                mmDevice = device;
                break;
            }
        }
    }

    myLabel.setText("Robot Tidak Ditemukan");
}

void openBT() throws IOException {
    UUID uuid = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    mmSocket = mmDevice.createRfcommSocketToServiceRecord(uuid);
}

```

```

mmSocket.connect();

mmOutputStream = mmSocket.getOutputStream();

mmInputStream = mmSocket.getInputStream();

beginListenForData();

myLabel.setText("Robot Ditemukan");
}

void beginListenForData()
{
    final Handler handler = new Handler();
    final byte delimiter = 10; //This is the ASCII code for a newline character

    stopWorker = false;
    readBufferPosition = 0;
    readBuffer = new byte[1024];
    workerThread = new Thread(new Runnable()
    {
        public void run()
        {
            while(!Thread.currentThread().isInterrupted() && !stopWorker)
            {
                try
                {
                    int bytesAvailable = mmInputStream.available();
                    if(bytesAvailable > 0)
                    {
                        byte[] packetBytes = new byte[bytesAvailable];
                        mmInputStream.read(packetBytes);

                        for(int i=0;i<bytesAvailable;i++)
                        {
                            byte b = packetBytes[i];
                            if(b == delimiter)

```



```

        {
            byte[] encodedBytes = new byte[readBufferPosition];
            System.arraycopy(readBuffer, 0, encodedBytes, 0, encodedBytes.length);
            final String data = new String(encodedBytes, "US-ASCII");
            readBufferPosition = 0;

            //Array push to Buffer
            bufferData.add(data);

            handler.post(new Runnable()
            {
                public void run()
                {
                    myLabel.setText(data);
                    for (int i=0;i<bufferData.size();i++){
                        arrBufferData.setText(bufferData.get(i));
                    }
                }
            });
        }
        else
        {
            readBuffer[readBufferPosition++] = b;
        }
    }
}

}
}
}
});

```

```

        workerThread.start();
    }

    void sendData() throws IOException {
        String msg = myTextbox.getText().toString();

        msg += "\n";
        mmOutputStream.write(msg.getBytes());
        myLabel.setText("Data Sent");
    }

    void closeBT() throws IOException {
        stopWorker = true;
        mmOutputStream.close();
        mmInputStream.close();
        mmSocket.close();
        myLabel.setText("Bluetooth closed");
    }

    public void startMap() {
        // TODO Auto-generated method stub
        super.init();

        if(android.os.Build.VERSION.SDK_INT
android.os.Build.VERSION_CODES.ICE_CREAM_SANDWICH_MR1) {
            fixJellyBeanIssues();
        }

        /* Menjalankan aplikasi web */
        super.loadUrl("file:///android_asset/www/index.html");
    }

    /* Fix issue */
    private void fixJellyBeanIssues() {
        // TODO Auto-generated method stub

```

```
System.out.println(super.appView.toString());

try {
    super.appView.getSettings().setAllowUniversalAccessFromFileURLs(true);
} catch(NullPointerException e) {
    System.out.println(e.toString());
}

}

@Override
public void onReceivedError( int errorCode, String description, String failingUrl)
{
    if(retryCount < 3) {
        retryCount++;
        System.out.println("Connection failed, trying again. Retry Count: "+retryCount);
        super.loadUrl("file:///android_asset/www/index.html");
    } else {
        System.out.println("Sorry, it failed three times so I give up.");
        super.loadUrl("file:///android_asset/www/fail.html");
    }
    return;
}
}
```

Lampiran 2

1. Kode Program I/O Writer Android

```
public class ProgressTask extends AsyncTask<Integer, Integer, Void> {
    @Override
    protected Void doInBackground(Integer... params) {
        // TODO Auto-generated method stub
        Writing();
        return null;
    }
    private void Writing() {
        // TODO Auto-generated method stub
        try {
            String[] lines = new String[888];
            File internalStorage = Environment.getDataDirectory();
            File dir = new File (internalStorage + "/myPde");
            File file = new File(dir, "ekfslam.pde");
            File file_temp = new File(dir, "ekfslam_temp.txt");

            FileWriter fw = new FileWriter(file_temp,true);
            BufferedWriter bw = new BufferedWriter(fw);
            LineNumberReader lnr = new LineNumberReader(new FileReader(file));
            file_temp.createNewFile();

            int i = 0;
            String Line = "";
            while (true) {
                Line = lnr.readLine();
                if (Line != null) {
                    lines[i]=Line;
                    i++;
                } else {
                    break;
                }
            }
            for (int j = 0; j < 50; j+=5) {
                lines[95] = lines[95]+"\\n\\t Data "+j;
            }

            if (file_temp.exists()) {
            } else {
                file_temp.createNewFile();
            }

            for (i = 0; i < lines.length; i++){
                bw.write(lines[i]);
                bw.newLine();
                System.out.println(lines[i]);
            }
            file.delete();
            file_temp.renameTo(file);
            bw.close();
            lnr.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```