

# Pengendali Gerbang Berbasis *Android*

Andrey Davied Ratulangi.<sup>(1)</sup>, Rizal Sengkey, ST, MT.<sup>(2)</sup>, Arie S.M. Lumenta, ST, MT.<sup>(3)</sup>  
 (1)Mahasiswa (2)Pembimbing 1 (3)Pembimbing 2

Jurusan Teknik Elektro-FT, UNSRAT, Manado-95115, Email: andrey4one@gmail.com

## Abstrak

Remot kontrol sangatlah membantu dan mempermudah aktivitas sehari-hari, sehingga ponsel pintar bisa dimodifikasi untuk pengendalian barang-barang elektronik guna kebutuhan manusia. Karena ponsel pintar *android* bersifat *open source* sehingga pengembang aplikasi ini dapat melakukan dengan bebas dan bisa di modifikasi. Hal ini menumbuhkan minat *developer software mobile* untuk dapat membuat perangkat lunak yang bermanfaat dalam kebutuhan manusia. Teknologi di dunia robotika saat ini sangat berkembang pesat dan semakin maju seiring perubahan waktu. Dalam sistem kendali permasalahan *interfacing* untuk mengontrol yang selalu dapat menjadi persoalan, sedangkan penggunaan kabel untuk mengontrol dipandang kurang efektif dan efisien.

Dalam Tugas Akhir ini bertujuan membuat aplikasi ponsel pintar berbasis *android* sebagai pengendali buka tutup gerbang dengan memanfaatkan *Bluetooth* dan menjadikan *android* sebagai perangkat selular yang multifungsi, di samping alat komunikasi tapi juga sebagai perangkat yang dikomunikasikan untuk mengendalikan sebuah perangkat keras.

Berdasarkan hasil pengujian yang dilakukan, sistem dapat bekerja dengan baik sesuai dengan tujuan *developer*. Aplikasi yang dibuat dapat dijalankan di berbagai jenis sistem operasi *Android*, dengan resolusi layar dan *hardware* yang berbeda-beda tiap perangkat *Android*. Komunikasi data robot dengan ponsel berjalan dengan baik dan data yang dikirim dapat diproses oleh gerbang secara *realtime* dengan waktu transfer kurang dari 1 detik pada kondisi sinyal yang berbeda-beda. Jarak jangkauan sistem aplikasi kendali dapat mengontrol gerbang tergantung spesifikasi perangkat *wireless* gerbang yang digunakan baik dari sisi penerima dan pengirim.

**Kata Kunci :** *Android*, Antarmuka, *Arduino*, *Bluetooth*, Pengembang Perangkat Lunak Ponsel, Pengendali

## Abstract

*Remote control is very helpful and simplify daily activities, so that smart phones can be modified to control electronic goods for human needs. Because the Android's smart phone is an open source, the application developers can modify it freely. This growing interest in mobile software makes the developers create a software that can be useful in human needs. Robotics technology in the world today is growing rapidly and increasingly advanced as time changes. In a control system for controlling the interfacing problems can always be a problem, while the use of cables to control is considered less effective and efficient.*

*This Final Project aims to create an Android-based smart phone applications as controlling the opening and closing of gates by utilizing Bluetooth and make the android as multifunctional mobile devices, not only as a communication tool but also as a device that is communicated to control a hardware device.*

*Based on the results of tests performed, the system can work well in accordance with the objectives developer. Applications that has been created can run on different types of Android operating system, with different screen resolution and hardware for each Android device. Data communication with a mobile robot goes well, and the data which has been sent can be processed by the gate in a realtime, with a transfer time of less than 1 second on a different signals condition. The range control system can control the gate application depends on specifications of the wireless gateway used, both from the recipient and the sender.*

**Key Words :** *Android, Arduino, Bluetooth, Control, Developer software mobile, Interfacing*

## I. PENDAHULUAN

Kemajuan dunia teknologi informasi yang begitu cepat ditunjang dengan penemuan dan inovasi telah membawa banyak perubahan dalam kehidupan manusia. Semakin banyak hal dan aspek dalam kehidupan yang menggunakan teknologi informasi untuk menjalankan roda aktivitasnya. Contoh pesatnya perkembangan teknologi informasi adalah perkembangan dari *smartphone* dan *mobile phone* yang semakin canggih yang telah menyebabkan perubahan besar pada kebiasaan pengguna kedua device tadi. Perubahan tersebut adalah penggunaan dari kedua device tadi yang pada awalnya hanya digunakan untuk keperluan telepon, pesan singkat, pesan elektronik dan browsing menuju pada penggunaan yang mulai menunjang kegiatan sehari-hari, baik itu untuk keperluan pekerjaan maupun keperluan hiburan. Hal ini berimbas pada meningkatnya kebutuhan user akan *smartphone*. Untuk mengatasi tingginya kebutuhan user akan *smartphone* maka dibutuhkan suatu mekanisme yang mudah untuk dapat mengakses *smartphone* secara mudah dan praktis, tanpa dibatasi oleh jaringan yang ada.

Untuk mengakses *smartphone* dengan mudah terutama *android*, tersedia banyak *software* di *android* market yang memudahkan user untuk mengakses perangkat *android* mereka dengan mudah, melalui jaringan Internet, *WiFi* ataupun *Bluetooth*. Namun biasanya aplikasi seperti ini terbatas pada jaringan lokal saja. Untuk aplikasi akses jarak jauh dibutuhkan koneksi internet dan server tersendiri dengan biaya yang tidak murah.

Berdasarkan pemahaman dari latar belakang ini sehingga penulis membuat tugas akhir dengan judul “*Pengendali Gerbang Berbasis Android*”. Dengan menggunakan Media Ponsel Pintar untuk membuka gerbang dan menutup gerbang, maka penulis membuat koneksi *Bluetooth* agar dapat melakukan koneksi remote pada perangkat *Android* ke *Microcontroller Arduino Uno R3*.

## II. LANDASAN TEORI

### *Ponsel Pintar*

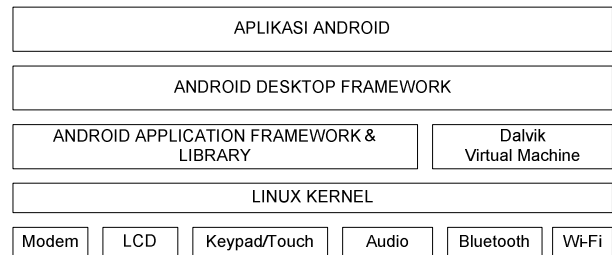
Ponsel pintar adalah perkembangan dari *handphone* yang ditambahi fitur - fitur seperti pada personal komputer fitur - fitur seperti *email*, *personal organizer*, dan juga konektivitas tambahan seperti *wifi* dan *bluetooth* yang dapat diinstall di *device*. Dari segi arsitektur *device* sendiri sudah dilengkapi dengan inputan seperti QWERTY miniatur *keyboard* dan *touchscreen*.

Ponsel pintar pertama diberi nama Simon yang dikembangkan oleh IBM pada tahun 1992 dan terpilih sebagai *product of the year* oleh COMDEX. Simon direlease pada tahun 1993 oleh BellSouth, selain fitur telephone dan SMS Simon dilengkapi dengan *calendar*, *address book*, *world clock*, *notepad*, *e-mail*, *fax*, dan *games*. Setelah itu banyak prodak sejenis yang dikeluarkan oleh berbagai *vendor* berbeda seperti Nokia 9000 (*communicater*), *Ericsson R3800*, dan lain-lain.

### *Android*

*Android* adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan *platform* yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, Google Inc. mengakuisi *Android Inc.* Yang mengembangkan *software* untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, yaitu konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. HTC Dream yang pertama memakai sistem operasi *Android*.

Dari segi arsitektur system pada gambar 1, *Android* merupakan sekumpulan *framework* dan *virtual machine* yang berjalan di atas kernel linux. *Virtual machine Android* bernama *Dalvik Virtual Machine (DVM)*, *engine* ini berfungsi untuk menginterpretasikan dan menghubungkan seluruh kode mesin yang digunakan oleh setiap aplikasi dengan kernel linux. Sementara untuk *framework* aplikasi sebagian besar dikembangkan oleh google dan sebagian yang lain dikembangkan oleh pihak ketiga (*developer*).



Gambar 1 Lapisan Arsitektur Sistem Operasi *Android* Secara Umum

Dengan dukungan *Software Development Kit (SDK)* dan *Application Programming Interface (API)* dari google memberikan kemudahan bagi pihak ketiga (*developer*) untuk membangun aplikasi yang dapat berjalan pada sistem operasi *Android*.

Selain itu terdapat metode baru dalam mengembangkan aplikasi di dalam sistem operasi *Android* menggunakan *Native Development Kit (NDK)*. *NDK* ini memungkinkan *developer* untuk mengembangkan aplikasi di dalam sistem operasi *Android* menggunakan bahasa pemrograman C atau C++.

### *Komponen Aplikasi Android*

Menurut King C. Ableson, *Android* memiliki empat komponen. Meliputi *activity*, *Broadcast Receiver*, *service* dan *content provider*. Komponen aplikasi dapat disebut juga sebagai elemen-elemen aplikasi yang bisa dikembangkan pada platform *Android*.

*Activity* merupakan bagian yang paling umum dari empat komponen *Android*. Suatu *activity* yang biasanya satu layar dalam aplikasi pengguna. Setiap *activity* diimplementasikan sebagai satu class yang memperluas dasar kelas *activity*.

Sebuah *content provider* mengatur sekumpulan data aplikasi yang terbagi (*shared*). Kita bisa menyimpan data di filesystem, sebuah database SQLite, di web, atau di metode penyimpanan data lainnya yang bisa diakses oleh aplikasi kita. Melalui *content provider*, aplikasi lain bisa memberikan query atau bahkan bisa memodifikasi, tentunya jika *content provider* mengizinkan aksesnya.

*Broadcast receiver* adalah komponen yang merespon terhadap siaran (*broadcast*) pengumuman yang dikeluarkan oleh sistem. Banyak siaran *broadcast* yang aslinya berasal dari sistem.

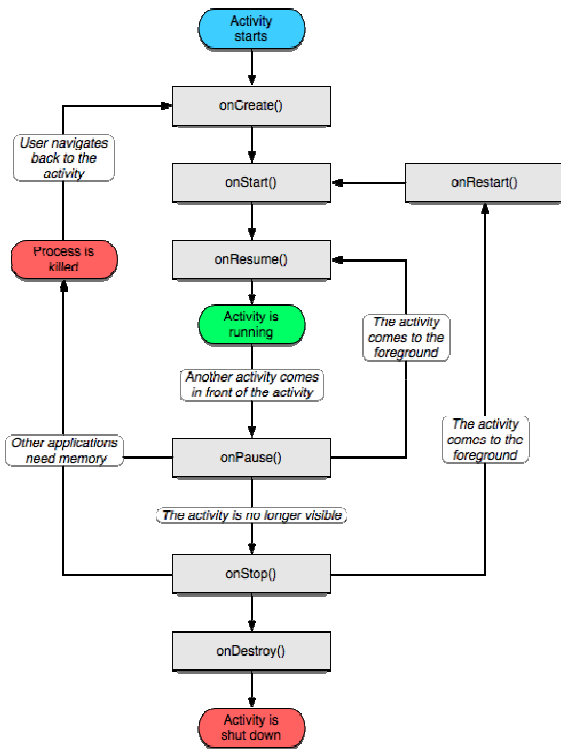
*Service* adalah komponen yang berjalan dibalik layar. Sebuah *service* tidak memiliki *user interface*.

### *Siklus Hidup Activity Android*

Komponen aplikasi *activity* diagram aliran program tampak pada gambar 2.

*onCreate*, dipanggil ketika *activity* pertama pada saat *onCreate* ini dijalankan akan menampilkan *layout* pada *background*.

*onStart*, dipanggil sebelum *activity* menampilkan *layout* pada layar perangkat seluler *Android*, ketika *onStart* berjalan maka *activity* pada sebuah aplikasi dapat berjalan dalam *foreground activity* yang nantinya dapat dipanggil oleh fungsi *onResume*.



Gambar 2 Diagram Alir Siklus Pemrograman Android

*onResume*, ketika ingin menampilkan *foreground activity* menjadi *background activity* maka fungsi ini lah yang digunakan untuk memanggilnya kembali menjadi *background activity*.

*onPause*, dipanggil ketika *activity* tidak lama akan terlihat karena *activity* lain akan berpindah ke *foreground activity*.

*onStop*, dipanggil ketika *activity* tidak lama ditutup kembali karena akan dijalankan pada *foreground activity*.

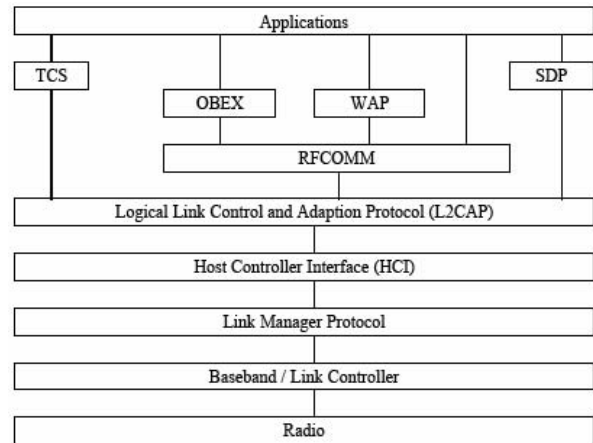
*onDestroy*, dipanggil untuk menghentikan seluruh proses *activity*.

### Bluetooth

*Bluetooth* tidak memerlukan daya yang besar untuk menghidupkannya atau memakainya, teknologi gelombang radio yang pendek yang dimaksudkan untuk menggantikan koneksi yang menggunakan kabel antar perangkat telepon genggam, PDA, dan perangkat lainnya. *Bluetooth* dapat digunakan untuk menghubungkan mouse, laptop, dan bagian komputer yang lainnya tanpa menggunakan kabel. Organisasi Ericsson Mobile Communications adalah pengembang *Bluetooth* pertama kali pada tahun 1994, dan organisasi tersebut bertujuan untuk mencari solusi untuk membuat koneksi antara telepon genggam dengan komponen yang ada (*accessories*) tanpa menggunakan kabel. Pada tahun 1999 perangkat *Bluetooth* mulai dipasarkan.

### Arsitektur Bluetooth

*Stack* dari *Bluetooth* terdiri dari beberapa layer, seperti tampak pada gambar 3, Layer HCI biasanya memisahkan antara perangkat keras dan perangkat lunak.



Gambar 3 Protocol Stack dari Bluetooth

### Piconet dan Scatternet

Sebuah *piconet* adalah jaringan *Bluetooth* yang terdiri dari satu *master* dan satu atau lebih *slave*. Suatu alat yang disebut *master* mengenali sebuah koneksi *Bluetooth* secara otomatis. Sebuah *piconet* dapat terdiri dari satu *master* dan tujuh *slave* yang aktif. Alat yang menjadi *master* secara harfiah menjadi *master* dalam suatu *piconet*. *Slave* hanya dapat mengirimkan data ketika waktu pengiriman diizinkan oleh alat yang pada saat tersebut menjadi *master*. *Slave* tidak boleh berkomunikasi langsung satu sama lain, semua proses komunikasi harus melewati *master*. *Slave* menyesuaikan frekuensinya dengan *master* menggunakan waktu *master* dan alamat *Bluetooth*.

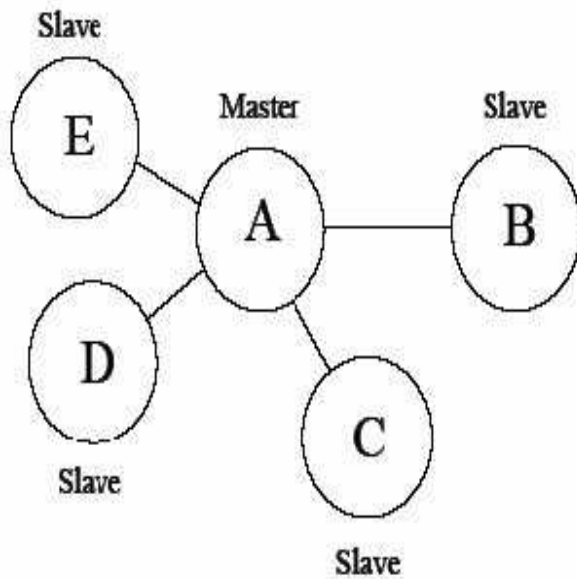
Konsep *piconet* diambil dari topologi jaringan yang sering disebut sebagai *star*, dengan sebuah *master* yang digambarkan sebagai *node* di tengah, ditunjukkan pada gambar 4. Dua *piconet* bisa terbentuk dengan jarak gelombang satu dengan yang lainnya. Frekuensi tidak sinkron antara *piconet* satu dengan *piconet* yang lain dan itu menyebabkan *piconet* yang lain secara acak akan bertabrakan pada frekuensi yang sama.

Dua *piconet* saling terhubung akan membentuk suatu jaringan yang sering disebut *scatternet*. Gambar 5 sebagai contohnya, dengan satu *node* di tengah yang menghubungkan *piconet* satu dengan *piconet* yang lainnya. *Node* yang berada di tengah harus membagi waktu, ini berarti *node* tersebut harus mengikuti perubahan frekuensi pada suatu *piconet* pada saat itu.

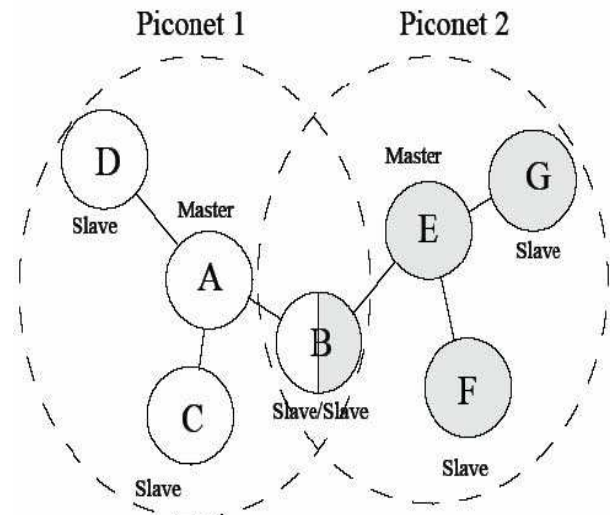
Jika anda memiliki dua alat A dan B. Alat A terhubung pada B, maka dari itu alat A akan menjadi *master* dari sebuah *piconet* yang terdiri dari alat A dan B.

Kemudian muncul alat yang bernama C ingin untuk bergabung dalam *piconet* tersebut. Alat C terhubung pada *master*, A. Sejak alat C dikenali oleh *piconet* tersebut, C secara otomatis akan menjadi *master* diantara alat A dan C. Kita sekarang memiliki dua *master*, sebab itu kita memiliki dua *piconet*. Alat A adalah alat dimana menjadi penengah antar *piconet-piconet* ini, alat A menjadi *master* untuk alat B dan menjadi *slave* untuk alat C.

Hubungan antara alat A dan alat C akan menghasilkan satu *piconet* dimana A sebagai *master* dan alat B dan C sebagai *slave*.



Gambar 4 Piconet



Gambar 5 Scatternet

### Keamanan Bluetooth

Keamanan merupakan faktor utama dan yang terpenting dalam berkomunikasi tanpa menggunakan kabel. Jika setiap alat dapat dikenali maka orang yang berada disekitar jangkauan *Bluetooth* dapat melakukan alat anda dengan menggunakan *Bluetooth*. Orang-orang tersebut dapat menentukan servis mana yang ditawarkan oleh alat itu dan mencoba untuk mengkoneksikannya. Masalah lainnya yang timbul adalah penyadapan paket yang sedang dikirim dan masalah itu akan dengan sangat mudah muncul karena data dikirim dari satu alat ke alat yang lainnya tanpa menggunakan kabel.

Untuk menghadapi ancaman-ancaman ini, *Bluetooth* dibagi menjadi tiga bagian yaitu pengautentifikasian, enkripsi data, dan pemberian hak akses. Sebagai tambahan tiga model keamanan dibuat dan lebih ditekankan pada level keamanan *Bluetooth* yang ke dua. Pengaturan keamanan digunakan untuk menangani transaksi keamanan pada sistem *Bluetooth*.

### Arduino

Untuk memahami Arduino, terlebih dahulu kita harus memahami terlebih dahulu apa yang dimaksud dengan *physical computing*. *Physical computing* adalah membuat sebuah sistem atau perangkat fisik dengan menggunakan software dan hardware yang sifatnya interaktif yaitu dapat menerima rangsangan dari lingkungan dan merespon balik. *Physical computing* adalah sebuah konsep untuk memahami hubungan yang manusiawi antara lingkungan yang sifat alaminya adalah analog dengan dunia digital. Pada prakteknya konsep ini diaplikasikan dalam desain alat atau proyek-proyek yang menggunakan sensor dan *microcontroller* untuk menerjemahkan input analog ke dalam sistem software untuk mengontrol gerakan alat-alat elektro-mekanik seperti lampu, motor dan sebagainya. Pembuatan *prototype* atau *prototyping* adalah kegiatan yang sangat penting di dalam proses *physical computing* karena pada tahap inilah seorang perancang melakukan eksperimen dan uji coba.

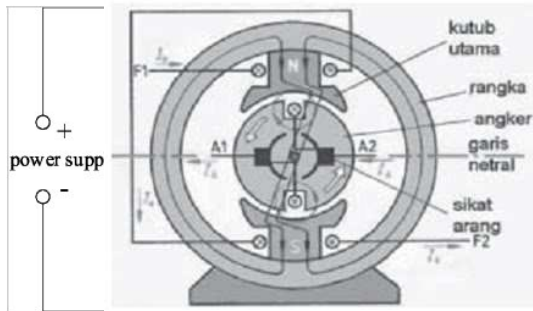
Dari berbagai jenis komponen, ukuran, parameter, program komputer dan sebagainya berulang-ulang kali sampai diperoleh kombinasi yang paling tepat. Dalam hal ini perhitungan angka-angka dan rumus yang akurat bukanlah satu-satunya faktor yang menjadi kunci sukses di dalam mendesain sebuah alat karena ada banyak faktor eksternal yang turut berperan, sehingga proses mencoba dan menemukan/mengoreksi kesalahan perlu melibatkan hal-hal yang sifatnya non-eksakta.

*Prototyping* adalah gabungan antara akurasi perhitungan dan seni. Proses *prototyping* bisa menjadi sebuah kegiatan yang menyenangkan atau menyebalkan, itu tergantung bagaimana kita melakukannya. Misalnya jika untuk mengganti sebuah komponen, merubah ukurannya atau merombak kerja sebuah *prototype* dibutuhkan usaha yang besar dan waktu yang lama, mungkin *prototyping* akan sangat melelahkan karena pekerjaan ini dapat dilakukan berulang-ulang sampai puluhan kali – bayangkan betapa frustasinya perancang yang harus melakukan itu. Idealnya sebuah *prototype* adalah sebuah sistem yang fleksibel dimana perancang bisa dengan mudah dan cepat melakukan perubahan-perubahan dan mencobanya lagi sehingga tenaga dan waktu tidak menjadi kendala berarti. Dengan demikian harus ada sebuah alat pengembangan yang membuat proses *prototyping* menjadi mudah.

*Arduino* dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata “*platform*” di sini adalah sebuah pilihan kata yang tepat. *Arduino* tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih.

### Motor DC Driver

Untuk aplikasi robot *mobile*, biasanya dapat digunakan beberapa aktuator. Salah satunya yang paling umum digunakan adalah motor listrik DC.



Gambar 6 Konstruksi Motor DC

Untuk aplikasi yang menggunakan motor listrik ini, kita hanya membutuhkan dua manipulasi pengendalian motor DC meliputi arah motor dan kecepatan dari motor.

Motor DC memiliki dua bagian dasar seperti pada gambar 6 yaitu bagian yang tetap/stasioner yang disebut *stator*. *Stator* ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen. Dan bagian yang berputar disebut *rotor*. *Rotor* ini berupa sebuah koil dimana arus listrik mengalir.

Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor.

### III. METODOLOGI PENELITIAN

Metodologi penelitian ini meliputi studi pustaka dan metodologi rekayasa perangkat lunak. Untuk studi pustaka diambil dari buku-buku dan referensi lain yang berhubungan dengan pokok bahasan. Adapun metodologi rekayasa perangkat lunak yang digunakan adalah *Classic life cycle* atau sering juga disebut metode *waterfall*. *Classic life cycle* adalah suatu paradigma perangkat lunak yang menuntut suatu sistem yang sistematis, mulai dari suatu level sistem kemudian terus maju ke level berikutnya. Terlepas dari segala kekurangannya model ini masih banyak yang menggunakannya dan dianggap tetap sesuai. Adapun tahapan *Classic life cycle* atau metode *waterfall* adalah sebagai berikut.

Analisis yaitu aplikasi yang akan dibuat merupakan aplikasi sistem kendali gerbang buka tutup otomatis yang diinstall dalam sistem operasi ponsel pintar *android* dengan jaringan *bluetooth*. Tentunya dalam pengembangan sistem aplikasi ini dibutuhkan objek gerbang yang akan dikendalikan dan perancangan *interface* antar perangkat keras yang digunakan, serta perangkat apa yang mendukung agar terjadinya komunikasi lewat *bluetooth* terbentuk.

Pada tahapan ini melakukan desain gerbang buka tutup otomatis sederhana, perancangan komunikasi data *bluetooth* dengan aplikasi *android* pengendali gerbang, komunikasi *bluetooth* dengan mikrokontroler, kemudian mikrokontroler dengan motor DC, serta desain yang paling urgen adalah *GUI* aplikasi *android* pengendali gerbang dan sistem pengendali melalui jaringan komputer berbasis *android*.

Penulisan program setelah tahap desain selesai, yaitu membuat aplikasi yang dapat digunakan di sistem operasi ponsel *android* untuk mengendalikan buka tutup gerbang.

Pada tahapan ini melakukan pengujian terhadap proses untuk eksekusi program yang telah selesai dibuat untuk memeriksa apakah terdapat kesalahan atau tidak. Pada tahapan ini terdapat beberapa segmen pengujian dari fungsional, kinerja sistem, *running* program di perangkat ponsel *android*, dsb.

Pada tahapan ini melakukan pemeliharaan dari program yang dibuat. *maintenance* dari aplikasi *android* ini memastikan dapat dijalankan disetiap versi sistem operasi *android*.

#### Tempat dan Waktu Penelitian

Tempat penelitian adalah Ruang Laboratorium Sistem Komputer (LSK), Jurusan Teknik Elektro, Fakultas Teknik Universitas Sam Ratulangi (UNSRAT)

#### Perancangan Sistem Pengendali Gerbang Secara Umum

Perancangan perangkat keras melalui tahapan, perancangan *interface Bluetooth* HC-05 dengan mikrokontroler dan perancangan *interface* mikrokontroler dengan motor DC seperti pada gambar 7.

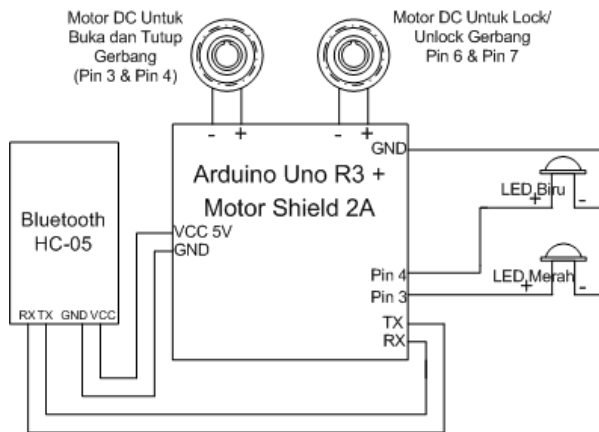
Pada tahapan perancangan perangkat lunak kita membuat suatu aplikasi *Android* sebagai *client socket* yang mampu mengirim perintah berupa *byte* ke *server socket*. Pada sisi *server socket* hanya mendengarkan dan membaca serta menunggu perintah baru dari *client server* jadi komunikasi terbentuk hanya *half duplex* tanpa adanya *feedback* dari sisi *socket server*. Pada tahapan ini juga mencakup perancangan program terhadap mikrokontroler *arduino* agar dapat membaca komunikasi serial port dari perangkat *bluetooth* dan sebagai aktuator motor DC pada gerbang.

HC-05 ini dapat digunakan sebagai *slave* maupun *master*. HC-05 memiliki 2 mode konfigurasi, yaitu *AT mode* dan *Communication mode*. *AT mode* berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *Communication mode* berfungsi untuk melakukan komunikasi *bluetooth* dengan piranti lain.

*Bluetooth* HC-05 ini memiliki koneksi slot 6 pin yang tersedia. Slot ini menyediakan dua port serial pada tegangan keluaran TTL 3.3 volt. Untuk menggunakan serial port ini kita hanya memfungsikan perintah *Receiver* dan *Transmitter* untuk mengatur RXD pada HC-05 ke mikrokontroler TX dan TXD pada HC-05 ke mikrokontroler RX. Modul ini akan mengkonversi sinyal tegangan *output* TTL 3,3 volt ke sinyal tegangan digital (0 dan 1) sebesar 5 volt.

Data yang terbentuk dari *Bluetooth* dengan *client Server* (rx,tx) kemudian diteruskan ke motor DC.





Gambar 7 Konfigurasi Sistem Hardware Pengendali Gerbang

Mikrokontroler sebagai otak untuk menyampaikan informasi digital *output* ke motor DC dengan perantara *Motor Shield 2A* untuk mengatur kecepatan dan arah putaran motor. Motor DC berperan sebagai Aktuator atau penggerak gerbang *bluetooth* ini.

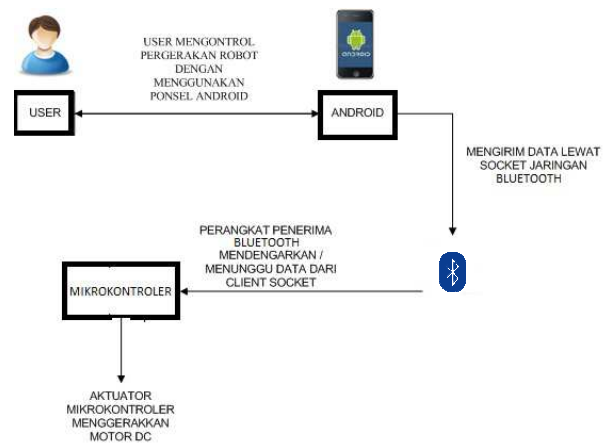
Program *Server* dibuat untuk mengetahui port yang akan digunakan dan memakai soket TX/RX *Server* untuk menerima koneksi ketika terjadi komunikasi data melalui pin *bluetooth*. *Server* ini menunggu data dari *client* untuk dieksekusi. Dalam hal ini *bluetooth* sudah ada standar protokol TCP yang tertanam di dalam alat untuk menerima koneksi protokol TCP (*Transmission Control Protocol*) ini karena protokol ini paling umum digunakan dalam pertukaran data suatu jaringan. Sehingga, port juga mengidentifikasi sebuah proses tertentu di mana sebuah *server* dapat memberikan sebuah layanan kepada *client* atau bagaimana sebuah *client* dapat mengakses sebuah layanan yang ada dalam *server*.

#### Desain Sistem Aplikasi Android Pengendali Gerbang

Desain aplikasi pengendali yaitu menggunakan papan tombol aplikasi *touch screen*. Maksud penulis menggunakan papan tombol sebagai *user interface* pengendali agar mudah dalam proses kendali dan mempunyai efektifitas dalam akses kendali. Aplikasi ini dibuat sebagai *client server*. Jadi inti dari algoritma aplikasi menggunakan *socket programming* di *android*. Aplikasi bertindak sebagai *client socket* aplikasi seperti pada gambar 8.

Aplikasi ini dibuat di *MIT App Inventor* atau bisa cek di internet [appinventor.mit.edu](http://appinventor.mit.edu). Aplikasi ini berbasis internet dan lebih mudah di gunakan, tidak perlu menginstal aplikasi-aplikasi pendukung lain. Aplikasi pengendali gerbang buka tutup *android*, menggunakan *bluetooth* menerapkan *socket programming* jaringan pada ponsel pintar *android*.

*ListPciker1* sebagai *coding* untuk membaca *bluetooth device*, dan setelah terhung dengan *device bluetooth* yang di pilih maka *BluetoothClient1* akan berubah satu menjadi *Connect* sehingga *Label1* akan memberikan *Text Connected* pada *label*. Dengan ini IP *Bluetooth* sudah secara default diatur di *hardware*, sehingga tidak perlu lagi untuk mengatur IP tetapi hanya memasukan *password* yaitu, 1234.



Gambar 8 Konfigurasi Sistem Aplikasi Pengendali Android

Dengan menekan tombol *GreenLedOpen* maka *bluetooth* akan mengirim sinyal 1 yang berarti membuka gerbang, tombol *GreenLedClose* maka *bluetooth* mengirimkan sinyal 2 yang berarti tutup gerbang, dan tombol *RedLedStop* untuk berhenti yang berarti sinyal 0 yang akan dikirimkan dari *bluetooth*.

## IV. HASIL DAN PEMBAHASAN

### Software

Dalam pembuatan aplikasi sistem kendali gerbang berbasis ponsel pintar *android* menggunakan program utama yaitu *MIT appinventor* berbasis online dan beberapa program penunjang yaitu *IDE arduino* sebagai *tool* untuk pemrograman mikrokontroler, serta ponsel dengan sistem operasi *android 4.2.2 (jelly bean)*.

### Hardware

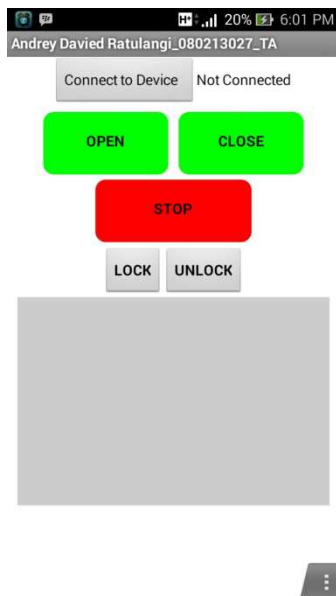
Selain *software* yang digunakan pada pembuatan sistem ini, juga digunakan *hardware* untuk mengoperasikan sistem ini. *Hardware* yang digunakan adalah mikrokontroler *arduino* dan *CD/DVD Rom* yang dimodifikasi sesuai kebutuhan sistem. Modifikasi tersebut meliputi penambahan *motor shield* dan perangkat *bluetooth*.

### Instalasi Program Aplikasi ke Perangkat Ponsel Android

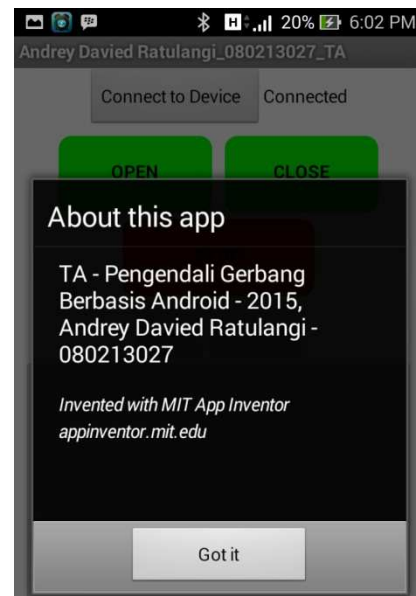
Proses instalasi dan kompilasi aplikasi *android* sistem pengendali dibuat di *MIT appinventor* berbasis online yaitu dengan mendesain langsung model *software* yang terdiri dari teknik *package file* dan *debugging* langsung ke perangkat ponsel berbasis *android* kemudian melalui emulator untuk hasil ujicoba aplikasi.

Instalasi langsung ke perangkat *android* terlebih dahulu melakukan desain aplikasi dengan menyertakan perancangan sistem kerja aplikasi. Program yang telah di buat akan mendapat ekstensi program *\*.apk (android package kit)*.

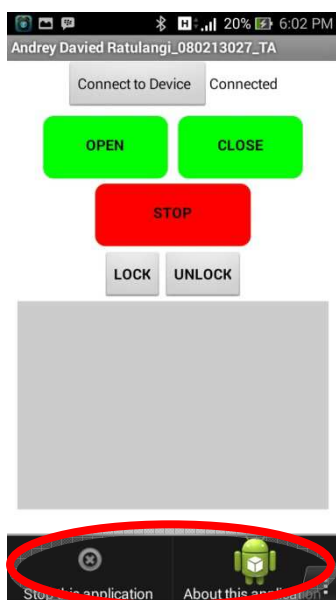
Pengecekan melalui *emulator android* adalah untuk mempermudah, bertujuan pengembang aplikasi dapat melihat hasil dari aplikasi yang dibuat tanpa perlu adanya perangkat ponsel *android*.



Gambar 9 Tampilan Utama Aplikasi androidDoorKey



Gambar 11 Tampilan Tentang/About androidDoorKey



Gambar 10 Tampilan Menu androidDoorKey

### Tampilan Program

Tampilan utama aplikasi ini terdiri dari *ListPicker* untuk memilih *access id* dan nama dari *bluetooth* untuk memulai koneksi dengan *server bluetooth*, *textView* untuk melihat status koneksi dan perintah *ImageButton/Label* untuk tombol buka, tutup, berhenti, kunci dan buka kunci seperti pada gambar 9.

Untuk koneksi port *bluetooth* ini hanya menggunakan standar dalam *bluetooth* yaitu 98:D3:31:B2:CB:6A dan nama HC-05. Karena *bluetooth* ini hanya bisa terkoneksi dengan aplikasi pengontrol pertama yang terhubung langsung di lokasi gerbang. Setelah memilih port yang akan di gunakan dengan tepat, maka *bluetooth* akan terkoneksi dan aplikasi siap untuk mengirim *Stream command* ke *server*.

Gambar 10 dan 11 merupakan antarmuka menu dari aplikasi *androidDoorKey*. ada 2 pilihan menu yaitu (*Stop this application*) / keluar dan (*About this application*) / tentang.

Fungsi dari menu (*About this application*) adalah menampilkan nama dan identitas aplikasi dari pembuat, sedangkan dari menu (*Stop this application*) adalah menutup/mengakhiri aplikasi.

### Pengujian Sistem

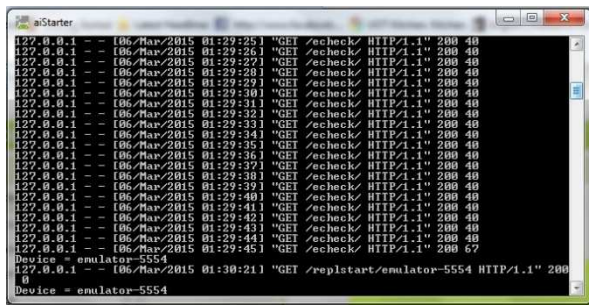
Pada Tahapan ini akan dibahas mengenai uji coba dan evaluasi perangkat lunak maupun perangkat keras. Uji coba perangkat keras meliputi uji coba dari gerak motor, konektivitas jaringan perangkat *bluetooth* pada gerbang. Sedangkan pada perangkat lunak, sistem dilakukan uji coba dari segi fungsionalitas dan performa dari sistem yang telah dibuat.

### Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk melihat dan menganalisa apakah fungsi-fungsi dasar dapat berjalan sebagaimana mestinya. Pengujian meliputi uji coba menjalankan sistem dan uji coba simulasi.

### Uji Coba Menjalankan Perangkat Lunak Sistem Kendali Android

Tahapan ini menguji program aplikasi *android* yang telah dibuat apakah dapat berjalan dengan berbagai jenis sistem ponsel *android* mulai dari *android 1.6 (Donut)*, *android 2.1 (Eclair)*, *android 2,2 (Froyo)*, *android 2.3 (Gingerbread)*, *android 3.0 (Honeycomb)*, *android 4.0 (Ice Cream Sandwich)*, *android 4.3 (Jelly Bean)*, *android 4.4 (KitKat)* sampai pada sistem *android* terbaru yaitu 5.0 (*Lollipop*). Sistem aplikasi dijalankan pada perangkat *android* yang mempunyai resolusi layar berbeda-beda resolusi layar, maupun spesifikasi *hardware*. Untuk efisiensi waktu dan fleksibilitas penulis melakukan uji coba menjalankan aplikasi sistem yang dibuat di emulator *android aiStarter* atau *AVD*



Gambar 12 Checking dan Instalasi Aplikasi di Emulator aiStarter

(*Android Virtual Device*) seperti pada gambar 12 dan 13. Aplikasi ini hanya memerlukan 324Kb dari RAM supaya dapat berjalan dengan baik di perangkat *android*.

Kompatibilitas dari sistem ini sangat diperhatikan agar setiap *device android* yang mendukung aplikasi dapat memanfaatkan aplikasi sistem pengendali ini sehingga sistem aplikasi yang dibangun bersifat juga *shareware* dan mudah dalam penggunaan dan pengembangan lebih lanjut.

#### Uji Coba Perangkat Lunak Program Socket Server Gerbang

Uji coba dilakukan untuk menjalankan program pada gerbang melalui mikrokontroler. Dalam hal ini program yang akan dikompilasikan pada perangkat *bluetooth* dan *motor shield* yaitu dengan *software IDE arduino*.

Sebelum program dikompilasi penulis memastikan serial port berfungsi dengan baik agar tidak ada *error* program.

Untuk proses kompilasi program, menggunakan modul *bluetooth* dan *motor shield* dengan cara memanipulasi dua *coding* modul menjadi satu *coding* untuk perintah pengdali gerbang.

Program *access* ini pada dasarnya adalah program berekstensi *UNIX* dan dikompilasi/dijalankan pada perangkat *wireless* gerbang *bluetooth* dengan perintah digital.

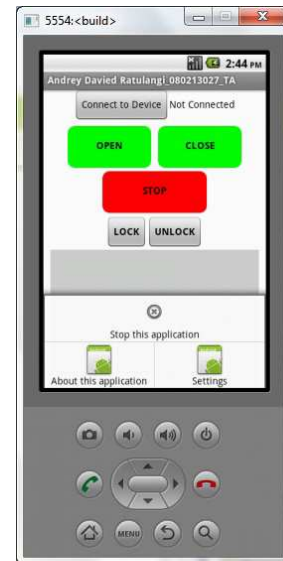
Program yang telah dikompilasi akan membentuk koneksi jaringan dengan perangkat *android* melalui alamat PORT dan NAMA dari *bluetooth* yang sudah ditentukan. Perintah yang dikirim dari ponsel *android* akan diterjemahkan dan diterima dari perangkat *wireless* gerbang *bluetooth* kemudian diteruskan ke mikrokontroler, dari mikrokontroler perintah kemudian dieksekusi untuk menjalankan gerbang yaitu motor DC.

Jangkauan jarak akses dari koneksi program, tergantung pada spesifikasi *hardware* yang mendukung perangkat *wireless* yang digunakan dalam hal ini *Bluetooth HC-05*.

#### Uji Coba Konektivitas Perangkat Lunak

Setelah sistem aplikasi pengendali gerbang dan program mikrokontroler dijalankan tahapan berikutnya menguji komunikasi antara perangkat lunak tersebut melalui koneksi jaringan *wireless*.

Pada ponsel *android* penulis mengaktifkan perangkat *bluetooth* kemudian dikoneksikan ke



Gambar 13 Aplikasi Dijalankan di aiStarter Android 2.2

*Access Point* gerbang *bluetooth* dan memasukan *password*. Ponsel akan terkoneksi dengan *device wireless* gerbang. Untuk mengetahui koneksi terbentuk atau tidak, yaitu dengan melihat pemberitahuan nyala led yang terdapat pada *bluetooth*. Sistem kendali program terkoneksi di PORT 98:D3:31:B2:CB:6A dan NAMA HC-05.

#### Uji Coba Jangkauan Bluetooth Pada Gerbang

Uji coba jangkauan *bluetooth* pada gerbang dilakukan untuk mengetahui jarak toleransi oleh sistem sehingga sistem dapat menerima data dengan baik dan mengetahui seberapa jauh jarak gerbang dapat dikendalikan melalui sistem yang dibuat.

Jarak efektif perangkat penerima pada umumnya *Bluetooth* bisa mengirim sinyal sampai 30 meter, namun perangkat penerima belum tentu bisa menjawab sinyal tersebut tergantung spesifikasi *hardware wireless* perangkat penerima tersebut. Secara teori jarak *Line-Of-Sight (LOS)* tanpa halangan adalah sekitar 25 meter, dan jarak Non LOS dengan halangan medium (didalam rumah) adalah sekitar 20 meter. Ini berlaku untuk 802.11 yang menggunakan frekuensi 2.4GHz.

#### Pengujian Pengiriman Data ke Gerbang

Pengujian selanjutnya yaitu komunikasi dengan mikrokontroler digerbang, dimana mikrokontroler akan memproses data yang dikirim dari perangkat *android* melalui *bluetooth* kemudian akan menggerakkan motor DC dan pada pengujian ini dapat mengetahui jarak toleransi pengiriman data ke gerbang.

Dalam keadaan kondisi didalam ruangan atau Non LOS pada tabel I jarak terdekat terbentuk koneksi *bluetooth* antara *client* dengan *Server* adalah 5 meter, kondisi sinyal *excellent* dalam 2 dBm dengan kondisi ruangan non LOS. Gerbang dengan aplikasi pengendali berjalan dengan baik walaupun jangkauan kendali di halangi tembok. Keadaan transfer data secara *realtime* tanpa ada *delay*.



TABEL I. PENGUJIAN PERTAMA PENGIRIMAN DATA DALAM KEADAAN DALAM RUANGAN

JARAK	KONDISI SINYAL
5 m	Excelent (2 dBm)
10 m	Medium (4 dBm)
15 m	Medium (5 dBm)
20 m	Low (5 dBm)
26 m	Low (6 dBm)

Kemudian untuk jarak 10 meter sampai 15 meter kondisi sinyal terlihat menurun status *bluetooth medium*, dalam 4 dBm sampai 5 dBm. Jarak jangkauan maksimum dalam hasil pengujian pada jarak 20 meter sampai 26 meter, dan kondisi status *bluetooth low*, karena kondisi sinyal menurun sehingga daya pancar pengiriman menjadi 5 dBm sampai 6 dBm. Hasil dari komunikasi transfer rata-rata antara gerbang dengan aplikasi yaitu berkisaran dibawah 1 detik.

Dalam keadaan kondisi diluar ruangan atau LOS pada tabel II ini memiliki hasil sinyal yang lebih baik daripada didalam ruangan, hal ini karena sinyal *transmitted bluetooth* terkondisi tanpa halangan. Pada jarak 10 meter kondisi sinyal masih *excelent* dalam 4 dBm dan jangkauan terjauh 30 meter, transfer data dalam *desibel* menjadi 7 dBm.

Hasil pengujian yang didapat dari tabel I dan tabel II bahwa jarak jangkauan kendali gerbang melalui *bluetooth*, data di transfer secara *realtime* dan berjalan dengan baik pada spesifikasi *bluetooth hc-05* penerima dan pengirim dengan *standard* 802.11.

#### Analisa Hasil

Kemampuan sistem kendali gerbang menggunakan ponsel pintar *android* dengan komunikasi data menggunakan jaringan *bluetooth* dalam penelitian ini berjalan dengan baik, aplikasi dapat digunakan diberbagai *platform* ponsel pintar yang mempunyai sistem operasi *android*. Dalam sistem pengontrolan menekan tombol pada aplikasi.

Aplikasi hanya dapat berfungsi apabila di gerbang *bluetooth* ini sudah terinstal *socket server* dan pada ponsel terinstal *client socket* aplikasi pengendali gerbang (*androiDoorKey*). Dari hasil pengujian jarak, kondisi tempat, dan spesifikasi perangkat *wirelles* yang digunakan sangat berpengaruh untuk terjadinya proses komunikasi data dengan baik. Jarak jangkauan kendali maksimum gerbang *bluetooth* pada kondisi didalam ruangan hanya mencapai 26 meter sedangkan pada kondisi diluar kondisi mencapai 30 meter.

Pada mode koneksi dengan *bluetooth*, data dapat terkirim dengan cepat dan bisa dibilang *realtime* pada kondisi sinyal *low*, *medium*, ataupun *excellent*. Jika di analisa lebih lanjut data yang dikirim berformat kode ASCII dengan ukuran masing-masing 1 byte. Sedangkan tipe *bluetooth* yang mendukung adalah v2.0 dengan data rate maksimum 3 Mb/s.

TABEL II. PENGUJIAN KEDUA PENGIRIMAN DATA DALAM KEADAAN LUAR RUANGAN

JARAK	KONDISI SINYAL
10 m	Excelent (4 dBm)
15 m	Excelent (5 dBm)
20 m	Medium (5 dBm)
25 m	Medium (5 dBm)
30 m	Low (7 dBm)

Dalam pembuatan sistem aplikasi pengendali ini ditemukan banyak hal yang dapat dikembangkan. Sinkronisasi komunikasi data merupakan persoalan tersendiri dari pemrograman sistem ini serta kemampuan membaca data dari komunikasi serial port perangkat *wireless* dengan mikrokontroler dan motor DC.

#### Masalah Dalam Sistem

Perancangan yang tepat akan meminimalisasi masalah yang akan ditemui tetapi tidak akan menghilangkan masalah. Itu pun terjadi pada penulis dalam pembuatan tugas akhir ini.

Membuat rancangan gerbang berbasis *android* tidaklah mudah walaupun didalam tugas akhir ini tidak membahas secara rinci dalam perancangan gerbang *bluetooth* ini karena terdapat berapa masalah dalam pengembangan perangkat *hardware*, salah satu adalah masalah durasi waktu kendali karena *supply* arus ke gerbang dan perangkat *wireless* yang digunakan pada tugas akhir ini terbatas sedangkan perangkat gerbang ini membutuhkan sumber yang lebih besar dan daya tahan *battery* lebih lama.

Tidak adanya aplikasi atau *software* yang bisa menghitung kecepatan pengiriman dan penerima data dari *bluetooth*, sehingga terbatasnya menghitung keakuratan waktu pengiriman dan penerima data dari *bluetooth*.

#### Penanganan Masalah

Untuk menyelesaikan tugas akhir ini, penulis mencari sumber data dan informasi dari banyak sumber baik secara tulisan maupun lisan. Bantuan dari mereka yang lebih menguasai rangkaian elektronika dan robotika membuat penulis mampu menyelesaikan pembuatan pengendali gerbang berbasis *android* meskipun masi ada kekurangan dalam perancangan *hardware*, namun penanganan masalah sistem bukan hanya pada perangkat keras tetapi juga pada sisi perangkat *software*. Penulis melakukan beberapa penyesuaian dengan mempelajari berbagai tutorial pengembangan aplikasi *android* dan cara pembuatan pengendali gerbang.

## V. KESIMPULAN

Setelah melalui proses perancangan dan pembuatan sistem ini, maka dengan memanfaatkan

ponsel pintar *android* sistem ini berhasil mengendalikan gerbang melalui koneksi *bluetooth*.

Kondisi sinyal pada *bluetooth* dalam *desibel* akan bertambah sesuai jarak jangkauan perangkat, apabila jarak semakin jauh maka *desibel* akan bertambah semakin besar sehingga mempengaruhi komunikasi transfer data pada aplikasi pengendali yang dibuat. Jarak jangkauan sinyal *bluetooth* tergantung pada spesifikasi dari perangkat *bluetooth* untuk bisa mengirim atau menerima data yang cepat, serta pada perancangan perangkat *hardware* yang digunakan sangat berpengaruh untuk komunikasi data transfer ke mikrokontroler sampai pada aktuator motor DC.

Dengan memanfaatkan teknologi pada perangkat *bluetooth HC-05*, dari hasil penelitian ini didapat bahwa perangkat ini dapat digunakan sebagai komunikasi serial dengan mikrokontroler.

#### DAFTAR PUSTAKA

- [1] D.D. Prasetya, *Membuat Aplikasi Smartphone Multiplatform*, Jakarta, 2013.
- [2] D. Suprianto, R. Agustina, S.Kom, M.Pd, *Pemrograman Aplikasi Android*, Yogyakarta, 2012.
- [3] D. Williams, *PDA Robotics*, McGraw-Hill, United States, 2003.
- [4] I.M. Siregar, S.T, M.T, *Membongkar Source Code Berbagai Aplikasi Android*, Yogyakarta, 2011.
- [5] R.L. Caratti, *Driving Stepper Motors With The L293D*, United States, 2014.
- [6] S.J.K. Egemen, *Introduction to Socket Programming*, tersedia di : [www.itc.ku.edu](http://www.itc.ku.edu), diakses tanggal 12 Desember 2011.
- [7] S. Nazruddin, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Penerbit Informatika Bandung, Bandung, 2011.