

Implementasi Sistem *Bluetooth* menggunakan Android dan Arduino untuk Kendali Peralatan Elektronik

PAULINE RAHMIATI¹, GINANJAR FIRDAUS², NUGRAHA FATHORRAHMAN¹

1. Jurusan Teknik Elektro Institut Teknologi Nasional Bandung

2. Nokia Siemens Network

Email: paulinebangun@gmail.com

ABSTRAK

Dalam suatu ruangan yang dipenuhi peralatan elektronik, remote control sangatlah penting. Keterbatasan sebuah remote control untuk mengendalikan hanya satu peralatan elektronik tentu menjadi kendala tersendiri. Untuk mengurangi kendala tersebut direalisasikan remote control yang dapat mengendalikan seluruh peralatan elektronik menggunakan perantara Bluetooth yang terintegrasi pada Android dan Arduino. Android akan mengirimkan perintah pada Arduino melalui Bluetooth, Arduino menerjemahkan perintah menjadi kode ke infra merah yang selanjutnya diterima oleh receiver peralatan elektronik. Dilakukan pengujian terhadap sistem remote control sehingga menghasilkan kesimpulan bahwa sistem remote control tersebut dapat berfungsi pada tiga versi Android berbeda, persentasi keberhasilan dari alat kendali 94,8% menggunakan tombol dan 92,8% menggunakan perintah suara, waktu akses rata-rata 0,17385 detik untuk perintah menggunakan tombol dan 0,20995 detik untuk perintah menggunakan perintah suara, dan jarak optimal pengirim infra merah (Tx) terhadap penerima infra merah (Rx) adalah 320 cm dengan lebar 180,8 cm.

Kata Kunci : Alat kendali, Bluetooth, Arduino, Android

ABSTRACT

In a room full of electronic equipment, control device are very important. The limitations of a remote control to control only one electronic device become obstacles that it needs a lot of control devices To reduce these constraints are realized remote control that can control the entire electronic devices using integrated Bluetooth intermediaries on Android and Arduino. Android will send commands to the Arduino via Bluetooth, Arduino code to translate the commands into infrared subsequently accepted by the receiver electronics. The first test installs Android application on three different versions. The second experiment tests all the tool buttons control for each electronic equipment. The third test is to examine the percentage of success of the control apparatus , producing 94.8 % success using the control buttons and 92.8 % control success using voice commands. The four test calculate the access time when sending a command until the electronic device executes the corresponding command. This test result in an average access time of 0.17385 seconds for commands using the buttons and 0.20995 seconds for command using voice control. The fifth testing is to test the coverage area of infrared , infrared sender optimal distance (Tx) to the infrared receiver (Rx) is 320 cm by 180.8 cm width

Keywords: remote control, Bluetooth, Arduino, Android

1. PENDAHULUAN

Dalam suatu ruangan yang dipenuhi peralatan elektronik, alat kendali sangatlah penting. Sebagian peralatan elektronik tidak memiliki alat kendali sehingga menyulitkan dalam mengontrolnya. Beberapa peralatan elektronik telah memiliki alat kendali untuk mempermudah penggunaannya tetapi terjadi kesulitan apabila ingin mengontrol lebih dari satu peralatan elektronik, ini dikarenakan *remote* kendali hanya dapat mengontrol satu peralatan elektronik saja. Semakin banyak peralatan elektronik yang akan dikontrol maka semakin banyak *remote* kendali yang dibutuhkan.

Untuk mengatasi masalah tersebut, perlu dirancangnya sistem untuk mengontrol seluruh peralatan elektronik dalam satu alat kendali. Alat kendali dapat memanfaatkan media komunikasi yang sekarang ini digunakan pada seluruh *smartphone* Android yaitu media komunikasi *Bluetooth*. Untuk itu, melalui integrasi dari *Bluetooth* pada perangkat *smartphone* Android dan *hardware* serta *software* yang digunakan, dihasilkan ide untuk merancang perangkat kendali peralatan elektronik berbasis *Bluetooth* pada *smartphone* Android.

Menurut Feri (**Feri Djuandi, 2011**) Arduino adalah sebuah *physical computing* yang bersifat *open source*. Dikutip dari artikel yang ditulis oleh Tim Diytech (**Tim DiyTech, 2012**) dijelaskan bahwa *Bluetooth* adalah teknologi komunikasi tanpa kabel yang menyediakan layanan komunikasi secara *real-time* antar perangkat *Bluetooth* dengan jarak layanan yang lebih jauh dari media infra merah. Teknologi *Bluetooth* banyak digunakan sebagai media pertukaran data pada berbagai perangkat *smartphone* termasuk Android. Tertulis pada artikel yang ditulis Tim Wikipedia (**Tim Wikipedia, 2008**) Android merupakan sistem operasi milik Google berbasis Linux yang dirancang untuk *smartphone* dan *tablet* dengan layar sentuh.

Melalui pengembangan dan perpaduan antara modul *Bluetooth* dengan Arduino dan infra merah dapat dirancang sebuah alat kendali berbasis *Bluetooth* pada Android untuk kendali perangkat elektronik yang mencakup seluruh fungsinya.

Penelitian yang dilakukan oleh Gema (**Gema Alfarisi Deri, 2013**) berjudul "Implementasi Sistem SMS *Gateway* untuk Kendali *Air Conditioner*", menjelaskan bahwa pada penelitian ini digunakan SMS yang memerlukan biaya dan pengendalian hanya terbatas pada *Air Conditioner* saja. Tujuannya adalah mengimplementasikan Sistem SMS *Gateway* untuk kendali *Air Conditioner* dengan berbasis Mikrokontroler ATMEGA 16. Bascom AVR digunakan sebagai *program* untuk menuliskan perintah-perintah atau *command* dan dimasukkan kedalam Mikrokontroler ATMEGA 16, untuk menjalankan perintah yang diterima dari *modem* GSM ke *remote universal* contohnya untuk menghidupkan atau mematikan AC. Kemudian setelah Mikrokontroler ATMEGA 16 menerima perintah maka akan mengirimkan informasi lagi ke *remote universal* yang kemudian dikirimkan lagi ke AC. Ada tiga pengujian yang dilakukan, yaitu pengujian pertama mengirim perintah SMS dengan *format* yang benar dan salah. Percobaan kedua mengirim sms dengan nomor yang tidak masuk dalam pengaturan. Pengujian ketiga menghitung lama waktu akses pada saat mengirim perintah hingga *user* menerima sms balasan bahwa AC telah aktif atau tidak. Dihasilkan waktu akses tercepat 6,756 detik dan waktu akses terlama 8,012 detik.

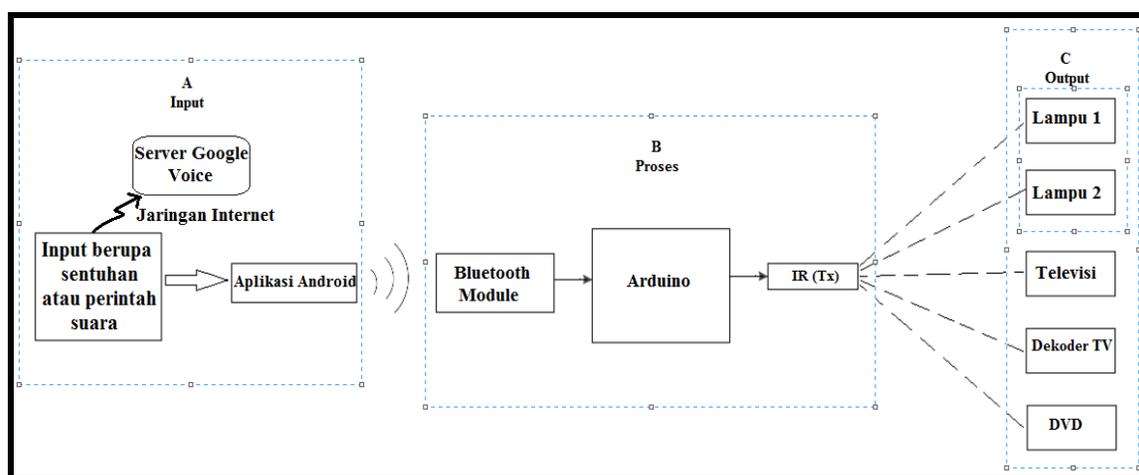
Penelitian yang dilakukan oleh Novi (**Noviyanti Tri Hapsari, 2012**) yang berjudul "Perencanaan Sistem Kendali Jarak Jauh Peralatan Listrik Rumah Tangga Dengan Kontrol *Wireless* menggunakan *Mobile Application* Berbasis Android", menjelaskan pengendalian hanya terbatas untuk mengaktifkan dan mematikan peralatan elektronik saja. Tujuannya adalah menciptakan suatu perangkat yang mampu digunakan untuk mengontrol piranti listrik

rumah tangga untuk menghidupkan/mematikan secara otomatis dan mampu dikendalikan pada jarak jauh dengan menggunakan Android *Mobile Application* via *Bluetooth*. Perangkat ponsel yang dipilih adalah ponsel yang memiliki system operasi Android yang kini sedang berkembang pesat dan aplikasi Android yang dibuat adalah pengembangan dari aplikasi *running text* yang sudah ada. *Single chip* ATMEGA 8 digunakan sebagai pengontrol untuk mematikan dan menghidupkan pada piranti listrik rumah tangga. *Bluetooth* bee sebagai modul *receiver* pada mikrokontroler.

2. METODOLOGI

2.1 Spesifikasi Perancangan Sistem

Secara garis besar, perancangan sistem digambarkan pada Gambar 1.

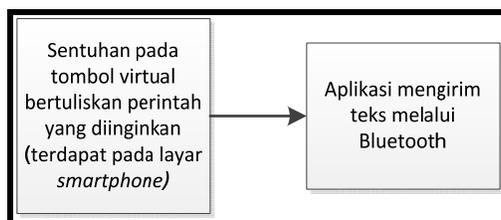


Gambar 1. Spesifikasi Perancangan Sistem

Seperti pada Gambar 1, sistem perancangan alat terbagi menjadi tiga yaitu *input* (A), proses (B), dan *Output* Tujuan akhir (C). Masing-masing bagian sistem akan dijelaskan sebagai berikut.

1. *Input*

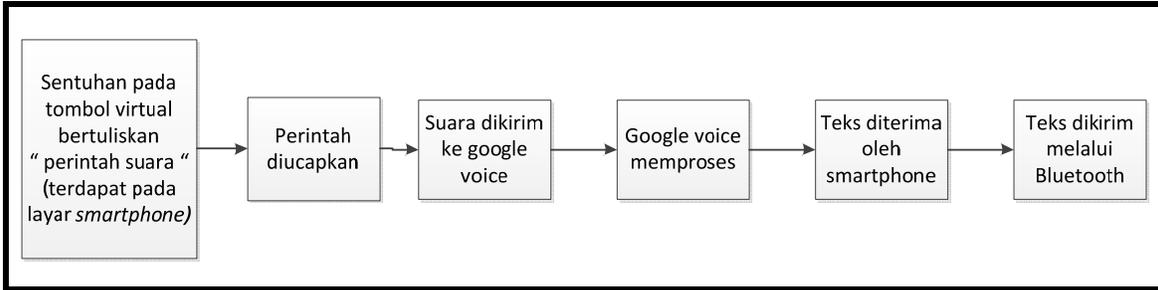
Input yang dimaksud adalah sebuah sentuhan atau perintah suara dari *user* yang kemudian direspon oleh aplikasi pada *smartphone* Android. Respon dari aplikasi Android ini berupa pengiriman sebuah teks melalui *Bluetooth*.



Gambar 2. Proses *Input* berupa Sentuhan pada Tombol Virtual

Gambar 2 merupakan proses *input* menggunakan sentuhan pada tombol virtual sesuai dengan perintah yang diinginkan. Masing-masing tombol virtual mewakili sebuah perintah

berupa teks yang siap dikirimkan setelah mendapat sentuhan. Contohnya ketika menyentuh tombol virtual bertuliskan menyala (lampu 1) maka aplikasi Android akan mengirimkan teks 1LON1 melalui *Bluetooth*.

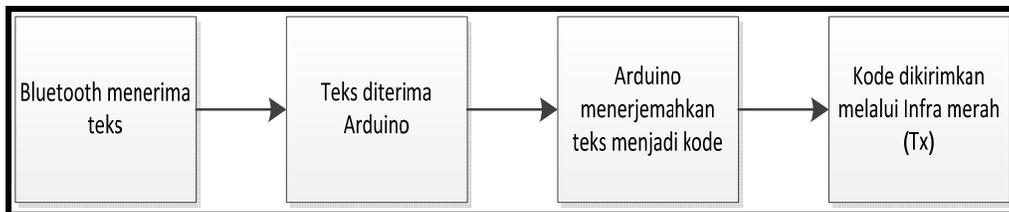


Gambar 3. Proses Input berupa Perintah Suara

Gambar 3 merupakan proses input menggunakan perintah suara. Langkah pertama adalah dengan melakukan sentuhan pada tombol virtual bertuliskan “perintah suara” sehingga muncul notifikasi aplikasi siap menerima perintah suara. Setelah itu perintah suara diucapkan oleh pengguna dan diterima melalui *microphone* yang terintegrasi pada *smartphone*. Suara tersebut dikirimkan ke Google voice melalui akses internet untuk diproses menjadi sebuah teks. Teks dari Google voice dikembalikan kembali melalui akses internet ke *smartphone*, kemudian aplikasi akan mencocokkan teks tersebut dengan basis data yang tersimpan. Apabila teks tersebut cocok dengan basis data yang tersimpan maka aplikasi akan mengirimkan perintah berupa teks yang sesuai melalui *Bluetooth*. Contohnya ketika mengucapkan “lampu menyala” maka aplikasi Android akan mengirimkan teks 1LON1 melalui *Bluetooth*.

2. Proses

Bagian proses merupakan bagian yang menerjemahkan teks dari *Bluetooth* menjadi kode untuk dikirimkan melalui infra merah.

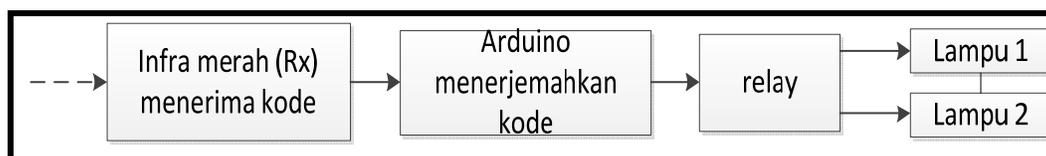


Gambar 4. Proses Menerjemahkan pada Sistem

Gambar 4 merupakan gambaran proses penerjemahan perintah teks menjadi kode. Teks diterima oleh *Bluetooth* melalui frekuensi radio. Teks tersebut dikirimkan *Bluetooth* ke Arduino melalui kabel. Arduino mencocokkan teks tersebut dengan basis data yang tersimpan. Apabila teks tersebut cocok dengan fungsi yang terdapat pada basis data, Arduino akan memberikan kode yang sesuai ke infra merah (Tx) yang tersambung melalui kabel. Selanjutnya infra merah mengirim kode ke tujuan dalam bentuk pancaran cahaya infra merah.

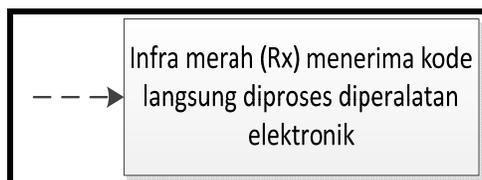
3. Output

Bagian Tujuan yang dimaksud adalah penerima akhir yang selanjutnya akan melakukan aksi yang diinginkan. Penerima ini merupakan peralatan elektronik yang ingin dikendalikan. Peralatan elektronik berdasarkan pengendaliannya dikategorikan menjadi dua bagian, yaitu peralatan elektronik yang telah memiliki *receiver* infra merah serta penerjemah sendiri dan peralatan elektronik yang belum memiliki *receiver* infra merah serta penerjemahnya. Untuk peralatan elektronik yang belum memiliki *receiver* infra merah serta penerjemahnya, memerlukan alat tambahan berupa sebuah sistem penerima yang terdiri dari *receiver* infra merah dan penerjemahnya. Dalam tugas akhir ini, sampel peralatan elektronik yang telah memiliki *receiver* infra merah serta penerjemahnya adalah Televisi, *Decoder* TV, dan DVD. Sedangkan sampel peralatan elektronik yang belum mempunyai *receiver* infra merah adalah dua buah lampu.



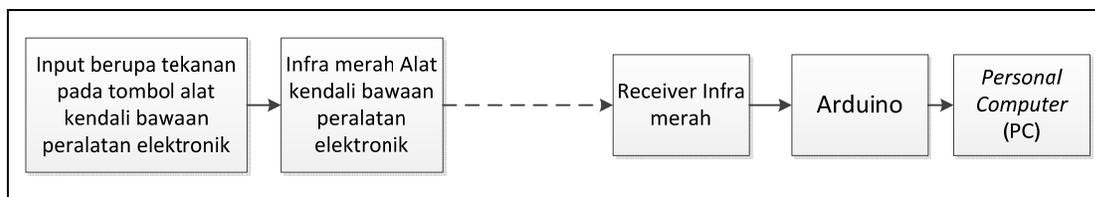
Gambar 5. Bagian Sistem Penerima Pada Lampu

Pada Gambar 5 merupakan ilustrasi sistem penerima infra merah yang terpisah dengan peralatan elektronik (dua buah lampu). Informasi cahaya infra merah yang disimbolkan dengan garis putus-putus diterima oleh *receiver* infra merah (Rx). Infra merah (Rx) meneruskan informasi tersebut dalam bentuk tegangan DC ke Arduino. Arduino memproses informasi tersebut menjadi sebuah kode yang akan dicocokkan dengan basis data. Jika data tersebut cocok maka Arduino akan memberikan atau memutuskan tegangan ke relay, sehingga relay akan melakukan aksi menyalakan atau mematikan lampu.



Gambar 6. Bagian Sistem Penerima Terintegrasi dengan Peralatan Elektronik

Sedangkan untuk perangkat elektronik yang memiliki sistem penerima infra merah yang terintegrasi (TV, *Decoder* TV, DVD), digambarkan pada Gambar 6. Pada kondisi ini tidak perlu memakai sistem penerima tambahan, tetapi diperlukannya proses penyalinan kode dari perangkat kendali bawaannya. Proses penyalinan ini digambarkan pada Gambar 7.

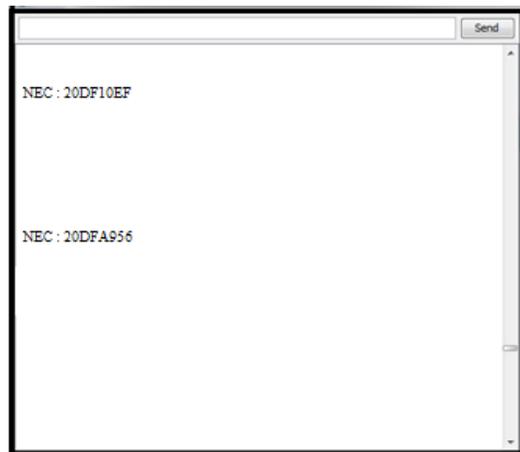


Gambar 7. Proses Penyalinan Kode Alat Kendali

Proses penyalinan ini digambarkan pada Gambar 7. Pembuatan sistem penyalinan alat kendali ini hanya bersifat sementara, karena Arduino tersebut akan digunakan untuk

pembuatan sistem lain. Cara kerja dari penyalinan alat kendali bawaan ini adalah sebagai berikut.

- a) Infra merah (Tx) alat kendali bawaan diposisikan sejajar dengan *receiver* infra merah yang terhubung dengan Arduino
- b) Tombol pada alat kendali ditekan sehingga alat kendali mengirimkan kode perintah berupa pancaran infra merah.
- c) *Receiver* infra merah menerima pancaran tersebut dan meneruskan informasi dalam bentuk tegangan DC ke Arduino.
- d) Arduino mengolah informasi tersebut menjadi kode dan ditampilkan pada layar PC dalam program *serial monitor*. Tampilan serial monitor ditunjukkan pada Gambar 8.



Gambar 8. *Serial Monitor* Pada Proses Penyalinan Kode

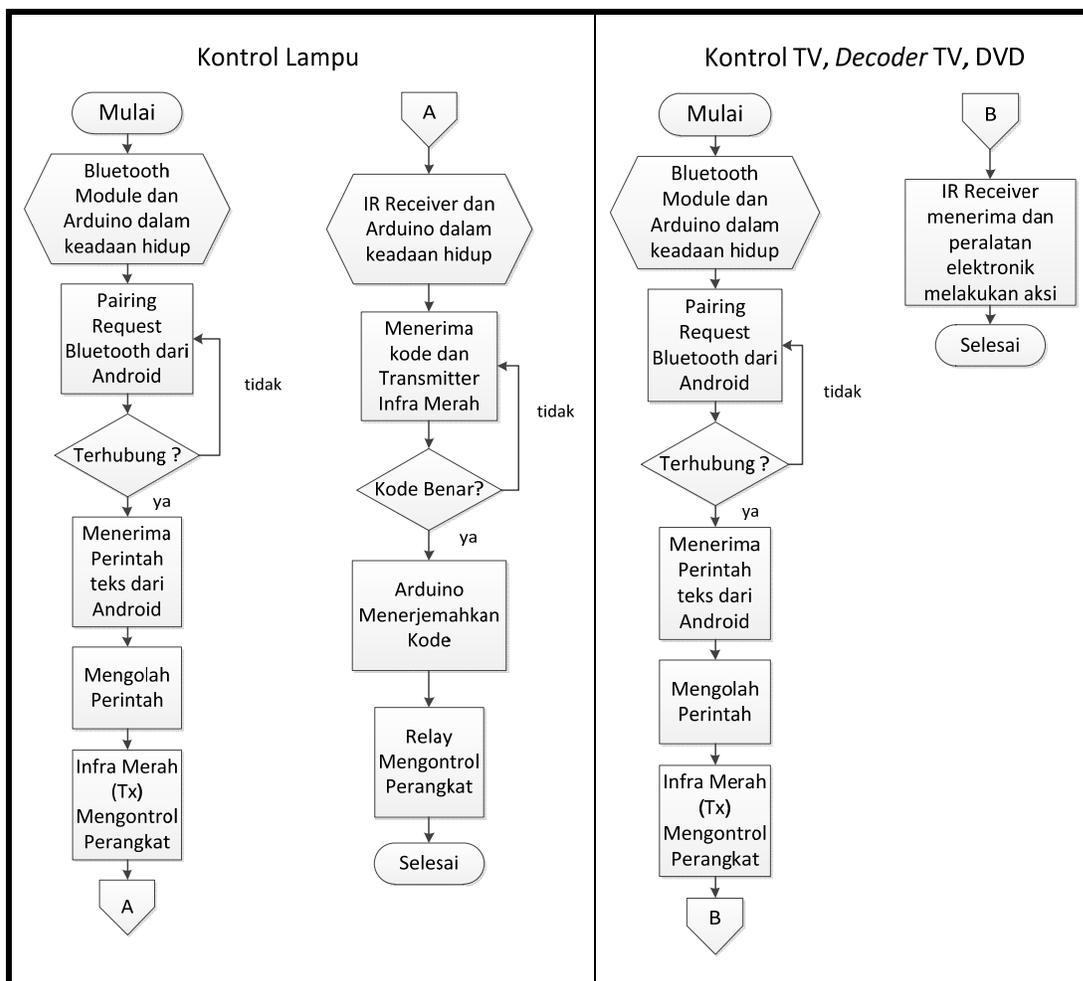
- e) Kode tersebut disalin untuk digunakan pada pembuatan *command* atau *sketch* pada Arduino, seperti pada Gambar 9.

```
if ( readString.indexOf("TVON")>0)
{
    irsend.sendNEC(0x20DF10EF,32);
    delay(100);
}
if ( readString.indexOf("TVSAV")>0)
{
    irsend.sendNEC(0x20DFA956,32);
    delay(100);
}
```

Gambar 9. Pembuatan Program Dari Proses Penyalinan Kode

- f) Step a hingga e diulang untuk tombol yang berbeda

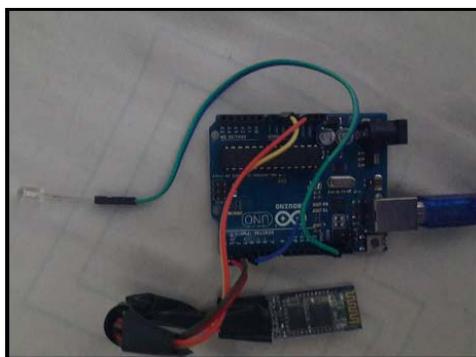
Untuk diagram kerja (*flow chart*) sistem pengontrolan dapat dilihat pada Gambar 10.



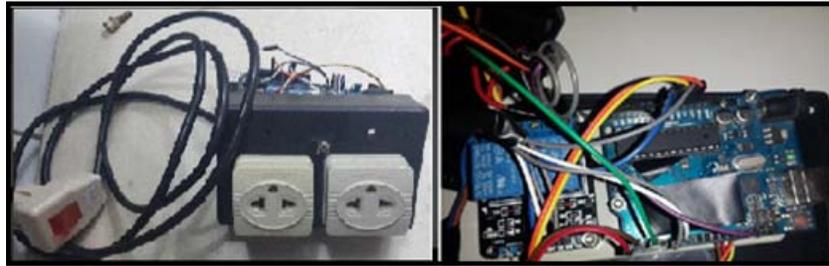
Gambar 10. Flow Chart Sistem Pengontrolan

2.2 Realisasi

Realisasi perangkat keras yang akan digunakan sebagai pemancar dan penerima dapat dilihat pada Gambar 11 untuk pengirim dan Gambar 12 untuk penerima.



Gambar 11. Realisasi Perangkat Keras Pengirim



Gambar 12. Realisasi Perangkat Keras Penerima Untuk Lampu

Adapun realisasi perangkat lunak (*software*) yang akan digunakan dapat dilihat pada Gambar 13 berikut ini.



Gambar 13. Realisasi Perangkat Keras Penerima Untuk Lampu

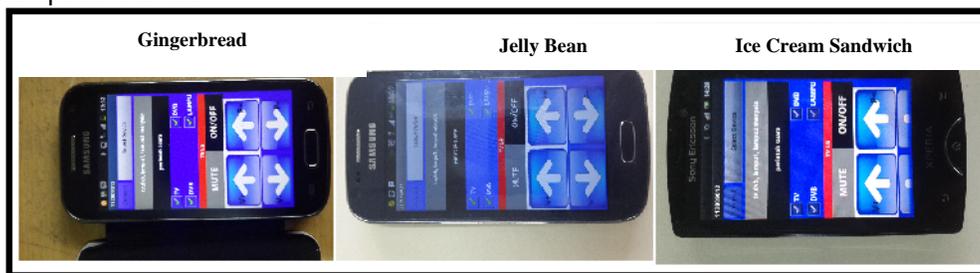
Untuk realisasi perangkat lunak yang dirancang terdiri dari 98 fungsi. 2 fungsi untuk mengatur sambungan ke *Bluetooth*, 1 fungsi untuk menghidupkan tiga peralatan elektronik secara bersamaan, 1 fungsi untuk perintah suara, 15 fungsi untuk mengatur TV, 15 fungsi untuk mengatur *Decoder TV*, 15 fungsi untuk mengatur DVD, 45 fungsi untuk *shortcut Decoder TV*, 2 fungsi untuk lampu pertama, dan 2 fungsi untuk lampu kedua.

3. PENGUJIAN DAN ANALISA

3.1 Pengujian Pemasangan Aplikasi pada Perangkat Android

Pengujian pertama adalah pengujian yang dilakukan untuk menguji kompetibelitas aplikasi yang dirancang terhadap perangkat Android dengan berbagai macam versi. Pengujian kompetibelitas pun dilakukan dengan cara memasang aplikasi yang telah dirancang pada perangkat Android dengan versi Gingerbread (GB), Ice Cream Sandwich (ICS), dan Jelly Bean (JB).

Setelah aplikasi terpasang selanjutnya, aplikasi tersebut dibuka sehingga menghasilkan tampilan pada Gambar 14.



Gambar 14. Tampilan Aplikasi yang Sedang Digunakan

Hasil dari pengujian aplikasi pada versi Gingerbread, Ice Cream Sandwich, dan Jelly Bean dapat dilihat pada Tabel 1.

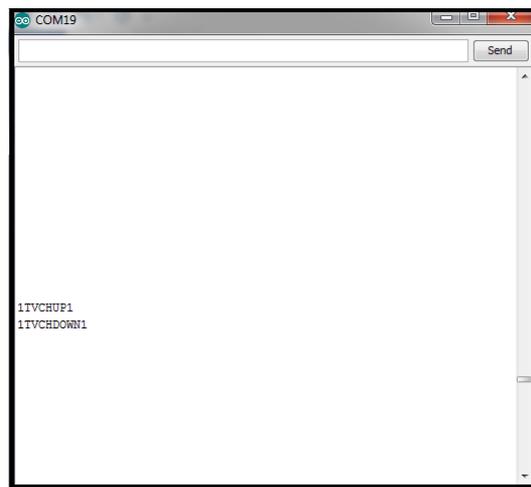
Tabel 1. Hasil Pengujian Aplikasi Diberbagai Versi

Nama Versi	Nama Perangkat	Jenis Pengujian		
		Pemasangan Aplikasi	Tampilkan Aplikasi	Tampilkan Daftar <i>Bluetooth</i>
Ginger Bread	X	Berhasil	Berhasil	Berhasil
Ice Cream Sandwich	Y	Berhasil	Berhasil	Berhasil
Jelly Bean	Z	Berhasil	Berhasil	Berhasil

Dari hasil pengujian pemasangan aplikasi ini, aplikasi yang dibuat dapat berjalan benar untuk semua versi sistem operasi Android. Ini dikarenakan aplikasi dibuat menggunakan Appinventor yang memang diperuntukkan bagi semua versi Android.

3.2 Pengujian Penerimaan Perintah Pada Modul *Bluetooth*

Pengujian ketiga adalah pengujian penerimaan perintah berupa teks pada mikrokontroler. Pengiriman perintah berupa teks dilakukan dengan menekan tombol pada aplikasi yang terpasang pada perangkat android. Hasil dari pengiriman tersebut dapat dilihat pada *serial monitor* IDE Arduino seperti Gambar 15.



Gambar 15. Tampilan Serial Monitor

Teks yang dikirimkan dari Android melalui *Bluetooth* dapat terbaca dengan benar pada *serial monitor* Arduino. Ini berarti sistem komunikasi *Bluetooth* antara Android dan Arduino berjalan dengan benar.

3.3 Pengujian Kendali TV, DVD, *Decoder* TV, dan Lampu

Pada pengujian mengendalikan TV, DVD, *Decoder* TV, dan kedua lampu ini dilakukan pada semua fungsi tombol. Hasil dari pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Tombol

Nama Perangkat	Jumlah Tombol	Jumlah Tombol yang gagal/ tidak sesuai fungsi
Televisi	15	0
<i>Decoder</i> TV	60	0
DVD	15	0
Lampu ke-1	2	0
Lampu ke-2	2	0

Seluruh tombol dapat bekerja secara benar untuk mengontrol TV, DVD, *Decoder* TV dan kedua lampu.

3.4 Persentasi Keberhasilan Kendali dengan Menggunakan Tombol

Pada pengujian ini dilakukan dengan menggunakan tombol untuk mengendalikan peralatan elektronik masing-masing sebanyak 100 kali pengujian. Hasil pengujian membuktikan persentasi keberhasilan sebesar 94,8% terlihat pada Tabel 3.

Tabel 3. Pengujian Persentasi Keberhasilan dengan Tombol

Jenis Perangkat	Tombol yang ditekan	Jumlah kegagalan
Televisi	V+	5
DVD	<i>Next</i>	7
<i>Decoder</i> TV	<i>Mute</i>	5
Lampu 1	Menyala/padam	4
Lampu 2	Menyala/padam	5

Terjadi total 26 kegagalan dalam 500 kali pengujian penekanan tombol, sehingga persentasi keberhasilannya adalah 94,8%. Kegagalan ini disebabkan keterlambatan respon Arduino dalam menerima perintah yang berulang. Keterlambatan Arduino untuk menerima perintah dikarenakan Arduino harus membaca basis data secara keseluruhan dan untuk menghindari *freeze* pada Arduino maka pada program diberi delay sebesar 1 detik apabila kondisi Arduino sibuk. Kesalahan tersebut tersebar secara *random* atau acak karena tidak memiliki pola tertentu. Dalam komunikasi *Bluetooth* tidak terjadi kesalahan penerimaan, ini terlihat dari tampilan *serial monitor*.

3.5 Persentasi Keberhasilan Kendali Menggunakan Perintah Suara

Pada pengujian ini dilakukan dengan menggunakan tombol untuk mengendalikan peralatan elektronik masing-masing sebanyak 100 kali pengujian. Hasil pengujian membuktikan persentasi keberhasilan sebesar 92,8% terlihat pada Tabel 4.

Tabel 4. Persentasi Keberhasilan dengan Menggunakan Perintah Suara

Jenis Perangkat	Perintah yang diucapkan	Jumlah kegagalan
Televisi	Tambah <i>volume</i> TV	8
DVD	<i>Film</i> berikutnya	6
<i>Decoder</i> TV	Suara <i>Decoder</i> hilang	9
Lampu 1	Lampu menyala/padam	7
Lampu 2	Lampu kedua menyala/padam	6

Terjadi total 36 kegagalan dalam 500 kali pengujian menggunakan perintah suara, sehingga persentasi keberhasilan adalah 92,8%. Kegagalan ini disebabkan oleh tiga hal, yaitu keterlambatan respon Arduino dalam menerima perintah yang berulang, akses jaringan internet buruk, dan kesalahan penerjemahan oleh Google voice. Penyebab kesalahan penerjemahan oleh Google voice ini dikarenakan pengucapan yang kurang jelas. Terdapat 2 kegagalan akibat buruknya akses jaringan internet, 4 kegagalan akibat kesalahan penerjemahan oleh Google voice, dan 30 kegagalan akibat keterlambatan respon Arduino. Semua kesalahan tersebut tersebar secara *random* atau acak karena tidak memiliki pola tertentu. Kegagalan menggunakan perintah suara lebih besar dibanding dengan menggunakan tombol karena terdapat tiga penyebab seperti yang telah dijelaskan. Selisih kegagalan perintah suara dibanding dengan menggunakan tombol adalah 10 kegagalan.

3.6 Pengujian Perhitungan Waktu Delay

Pengujian waktu delay ini dilakukan dengan cara menghitung waktu dari mulai perintah diberikan oleh Android hingga mikrokontroler mengeksekusi perintah lalu mengirimkan teks kembali ke Android. Proses menghitung waktu dari mulai perintah diberikan oleh Android hingga dikembalikan kembali oleh mikrokontroler disebut waktu keseluruhan.

Waktu delay sistem adalah menghitung waktu dari mulai perintah diberikan oleh Android hingga perintah tersebut dieksekusi oleh mikrokontroler. Sehingga untuk menghitung waktu delay dalam sistem adalah :

$$\text{Waktu delay sistem} = \frac{\text{Waktu keseluruhan}}{2} \quad (1)$$

3.6.1 Pengujian Menggunakan Tombol

Untuk pengujian ini dilakukan dengan menekan salah satu tombol pada aplikasi yaitu ON/OFF (TV). Pengujian dilakukan 10 kali percobaan dan didapatkan waktu respon rata-rata secara keseluruhan seperti pada Tabel 5.

Tabel 5. Pengujian Waktu Delay Menggunakan Tombol

Tombol yang Ditekan	Teks yang dikirim	Waktu delay keseluruhan (detik)	Waktu delay sistem (detik)	Waktu delay keseluruhan rata-rata (detik)	Waktu sistem rata-rata (detik)
ON (TV)	1TVON1	0.354	0.1770	0.3447	0.17385
		0.336	0.1830		
		0.350	0.1750		
		0.347	0.1735		
		0.341	0.1705		
		0.349	0.1745		
		0.350	0.1750		
		0.344	0.1720		
		0.340	0.1700		
		0.336	0.1680		

Waktu delay sistem merupakan setengah dari waktu delay keseluruhan, ini dikarenakan waktu delay sistem merupakan setengah dari proses kerja pada waktu delay keseluruhan. Rata-rata nilai delay sistem adalah 173,85 mili detik sehingga tidak terasa adanya jeda.

3.6.2 Pengujian Menggunakan Perintah Suara

Untuk pengujian ini dilakukan dengan memberikan salah satu perintah suara pada aplikasi yaitu "Televisi hidup". Pengujian dilakukan 10 kali percobaan dan didapatkan waktu respon rata-rata seperti pada Tabel 6.

Tabel 6. Pengujian Waktu Delay Dengan Perintah Suara

Perintah suara yang diucapkan	Teks yang dikirim	Waktu delay keseluruhan (detik)	Waktu delay sistem (detik)	Waktu Respon rata-rata (detik)	Waktu sistem rata-rata (detik)
Televisi hidup	1TVON1	0.412	0.2060	0.4191	0.20995
		0.408	0.2040		
		0.411	0.2055		
		0.409	0.2045		
		0.439	0.2195		
		0.456	0.2280		
		0.409	0.2045		
		0.410	0.2050		
		0.414	0.2070		
		0.423	0.2115		

Waktu delay sistem merupakan setengah dari waktu delay keseluruhan, ini dikarenakan waktu delay sistem merupakan setengah dari proses kerja pada waktu delay keseluruhan. Rata-rata nilai delay sistem adalah 209,95 mili detik delay ini lebih besar bila dibandingkan pengontrolan dengan tombol (173,86 mili detik). Selisih antara pengontrolan suara dan pengontrolan dengan tombol adalah 36,09 mili detik. Walaupun dengan *delay* yang lebih besar tidak terasa adanya jeda.

3.7 Pengujian Cakupan Infra Merah

Pengujian cakupan infra merah dilakukan dengan memindahkan posisi *receiver* infra merah setiap jarak 20 cm. Digunakan *receiver* infra merah pada lampu untuk pengujian ini.

Posisi *receiver* infra merah digeser hingga *receiver* infra merah tidak bisa menerima perintah dari *transmitter*. Selain itu jarak *receiver* diubah setiap 20 cm hingga *receiver* tidak dapat menerima perintah dari *transmitter*.

Hasil dari pengujian cakupan ini dapat dilihat pada Tabel 7. Pengujian dilakukan sebanyak 25 kali. Jarak terjauh yang didapatkan adalah 440 cm. Setelah jarak 340 cm, lebar cakupan infra merah semakin berkurang hingga akhirnya hilang pada pengujian ke-23. Pada tabel terlihat perbandingan jarak terhadap lebar cakupan infra merah satu sisi sedangkan pada pengujian ini, yang dibutuhkan adalah lebar cakupan infra merah keseluruhan pada dua sisi.

Tabel 7. Pengujian Jarak dan Lebar Cakupan Infra merah (Tx)

Pengujian ke-	Jarak "x" (cm)	Lebar "y" (cm)
1	20	24
2	40	41.8
3	60	32.3
4	80	34
5	100	36.8
6	120	43.5
7	140	47.1
8	160	54.1
9	180	56.8
10	200	62.2
11	220	69
12	240	69.2
13	260	71.4
14	280	76.6
15	300	80.7
16	320	85.3
17	340	90.4
18	360	69.5
19	380	65.4
20	400	43.4
21	420	35.6
22	440	12.3
23	460	-
24	480	-
25	500	-

Untuk lebar cakupan sebenarnya adalah dua kali nilai lebar "y" tersebut. Pada tabel terlihat jarak optimal terjauh 340 cm. Dikatakan memiliki jarak optimal terjauh karena pada jarak 430 cm didapat lebar terjauh, yaitu 180.8 cm.

4. KESIMPULAN

Dari hasil pengujian dan analisa dapat ditarik kesimpulan bahwa:

1. Pada pengujian untuk menghidupkan televisi dengan tombol dan perintah suara, didapatkan waktu respon rata-rata sebesar 0.17385 detik untuk penggunaan tombol dan 0.20995 detik untuk perintah suara. Perbedaan ini disebabkan pada perintah suara membutuhkan akses jaringan internet untuk terhubung dengan *google voice* sehingga memerlukan waktu yang lebih lama.
2. Pengujian persentasi keberhasilan dilakukan sebanyak 500 kali. Pengujian menggunakan tombol didapatkan persentasi keberhasilan 94.8 % sedangkan untuk menggunakan perintah suara didapatkan persentasi keberhasilan 92.8% .
3. Aplikasi dapat berjalan dengan benar di tiga perangkat Android yang berbeda mulai dari pemasangan aplikasi hingga penggunaan aplikasi. (indikator perbedaan berdasarkan pada versi sistem operasi, ukuran layar, *vendor*, prosesor, *graphic processing unit*, dan versi *Bluetooth*).
4. Jarak optimal pengirim infra merah (Tx) terhadap penerima infra merah (Rx) adalah 320 cm dengan lebar 180.4 cm.

DAFTAR RUJUKAN

- Djuandi, Feri (2011). *Pengenalan Arduino*. Dipetik November 8, 2013 dari <http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>
- Tim DiyTech (2012). *Bluetooth Module*. Dipetik December 14, 2013 dari <http://diytech.net/2012/03/07/dalam-beberapa-aplikasi-atau-disain-kadangkala-kita-memerlukan/>
- Tim Wikipedia (2008). *Android (Sistem Operasi)*. Dipetik December 14, 2013 dari [http://id.wikipedia.org/wiki/Android_\(sistem_operasi\)](http://id.wikipedia.org/wiki/Android_(sistem_operasi))
- Hapsari, Noviyanti Tri. (2012). *Perencanaan Sistem Kendali Jarak Jauh Peralatan Listrik Rumah Tangga Dengan Kontrol Wireless Pada Orange House Menggunakan Mobile Application Berbasis Android*. Bandung: Institut Teknologi Telkom.
- Deri, Gema Alfarisi. (2013). *Implementasi Sistem SMS Gateway untuk Kendali Air Conditioner*. Bandung: Institut Teknologi Nasional.