

BAB IV

PENGUJIAN DAN EVALUASI

Pengujian sistem yang dilakukan merupakan pengujian terhadap peralatan dan aplikasi yang telah dibuat. Pengujian tersebut dimulai dari pengujian aplikasi kendali perangkat elektronik yang nantinya secara otomatis melakukan pengujian lampu dan pengujian pagar.

4.1 Pengujian Aplikasi Kendali Perangkat Elektronik

Pengujian Aplikasi Kendali Perangkat Elektronik dari app inventor ke *smartphone* dilakukan dengan menguji kinerja app inventor sebagai aplikasi yang akan mempermudah dalam membuat aplikasi *smartphone*.

4.1.1 Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui apakah simulasi yang dibangun sudah berjalan sesuai dengan harapan yaitu membuat posisi *mobile node* secara *random*.

4.1.2 Peralatan Yang Digunakan

1. Perangkat keras (*Hardware*)
 - a. Laptop
 - b. Modem
 - c. *Smartphone* android
2. Perangkat Lunak (*Software*)
 - a. App Inventor

4.1.3 Prosedur Pengujian

1. Nyalakan laptop hubungkan, komputer dengan modem dengan koneksi *wifi*.

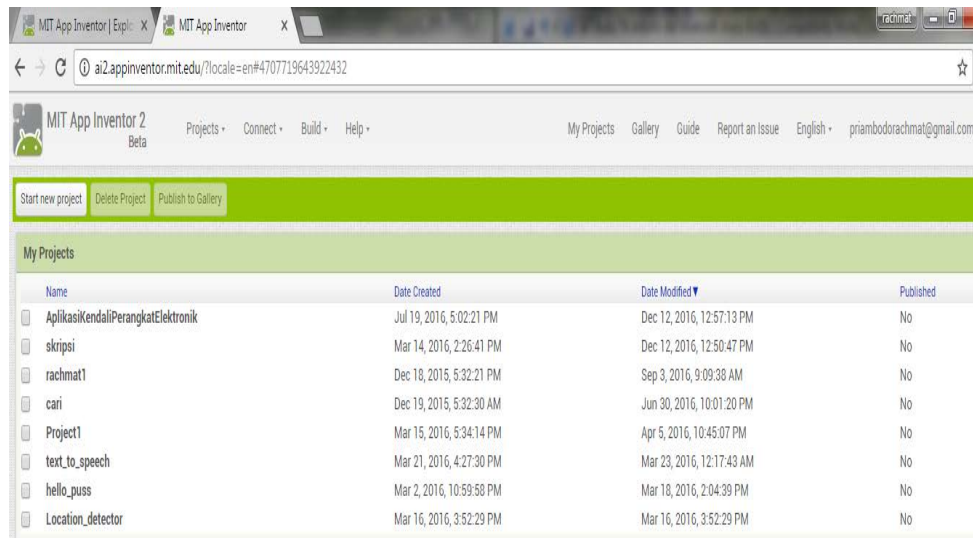
2. Buka *web browser* pada laptop dan masukkan alamat appinventor.mit.edu untuk membuka aplikasi *online* dari app inventor. Lalu pilih menu “*create apps!*”, masukkan id gmail, lalu pilih menu yang bernama “AplikasiKendaliPerangkatElektronik” pada kolom *my project*.
3. Ubah aplikasi menjadi format *.apk* dengan memulih menu “*build*” lalu pilih “App (*save .apk to my computer*)”.
4. Setelah aplikasi tersimpan di laptop, lakukan proses penginstalan aplikasi yang telah dibuat dengan memanfaatkan *ftp mode* pada *smartphone* oppo.
5. Aktifkan aplikasi kendali perangkat elektronik.

4.1.4 Hasil Pengujian Aplikasi Kendali Perangkat Elektronik

Pengujian aplikasi kendali perangkat elektronik ini digunakan untuk memastikan bahwa aplikasi yang dibuat dengan app inventor dapat diinstal pada *smartphone* android. Gambar 4.1 merupakan halaman awal appinventor.mit.edu

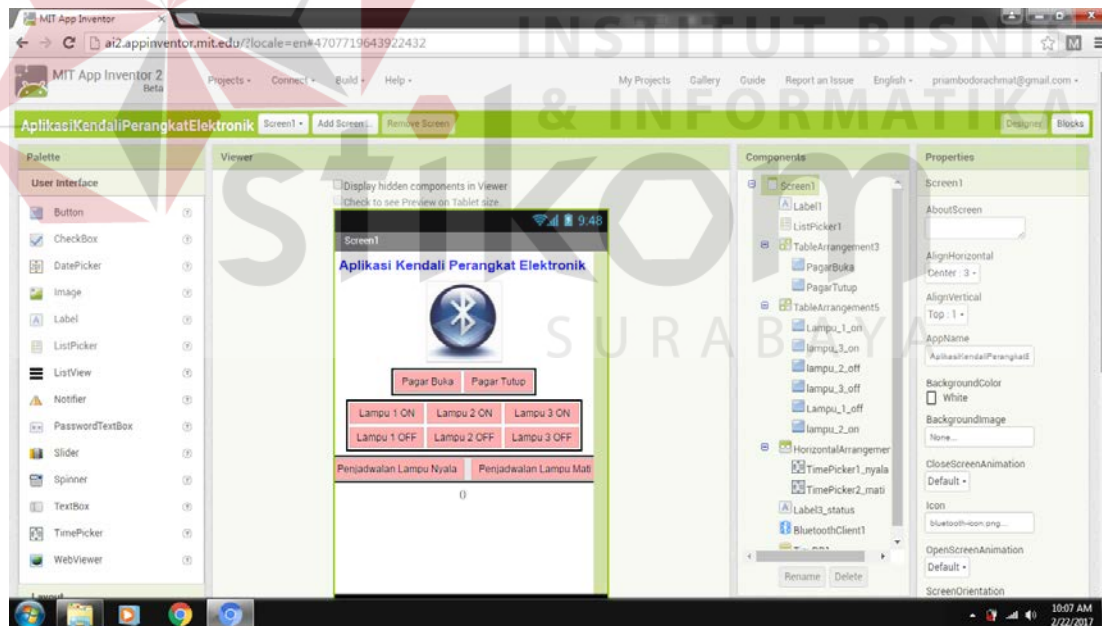


Gambar 4.1 tampilan awal appinventor.mit.edu



Gambar 4.2 tampilan *my project* app inventor

Setelah *login* dan memasukkan akun gmail pada *appinventor.mit.edu*, maka, tampilan *my project* akan ditampilkan pada Gambar 4.2.

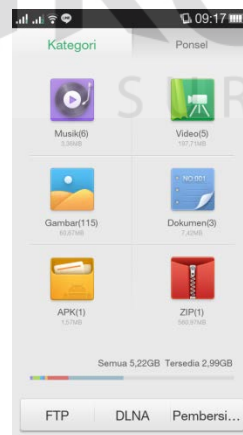


Gambar 4.3 tampilan AplikasiKendaliPerangkatElektronik

Jadikan format .apk dengan memilih menu “*build*” lalu pilih “App (*save .apk to my computer*)” seperti yang ditunjukkan pada gambar 4.4. Setelah aplikasi tersimpan di laptop, proses penginstalan aplikasi yang telah dibuat dengan memanfaatkan *ftp mode* pada *smartphone* oppo dijelaskan dengan gambar 4.5 dan gambar 4.6.



Gambar 4.4 menu *build* pada app inventor

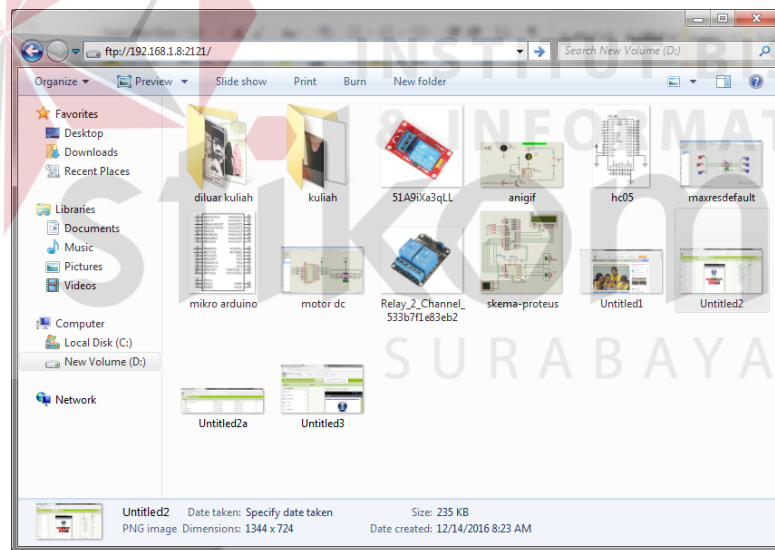


Gambar 4.5 *ftp mode* pada oppo *smartphone* 1



Gambar 4.6 ftp mode pada *smartphone* oppo 2

Untuk memulai ftp mode, maka lakukan pengisian alamat ftp://192.168.1.8:2121 sehingga proses pengiriman data dapat dilakukan seperti pada gambar 4.4.



Gambar 4.7 pengisian alamat ftp://192.168.1.8:2121

Aplikasi dapat di instal pada *smartphone* seperti yang tertera pada gambar 4.8, sehingga aplikasi kendali perangkat elektronik siap digunakan untuk menggerakkan pagar dan lampu otomatis.



Gambar 4.8 aplikasi berhasil diinstal

4.2 Pengujian Kontrol Lampu

Pengujian lampu dilakukan dengan menguji apakah komunikasi antara aplikasi dan rangkaian lampu telah berfungsi dengan baik.

4.2.1 Tujuan

Tujuan dari pengujian ini adalah untuk menguji apakah lampu akan menyala atau mati setelah dikontrol melalui aplikasi kendali perangkat elektronik.

4.2.2 Peralatan Yang Digunakan

1. Perangkat keras (*Hardware*)
 - a. *Smartphone* android
 - b. Arduino Uno
 - c. *Bluetooth* HC-05
 - d. Lampu
 - e. *Single Board Relay*

2. Perangkat Lunak (*Software*)

- a. Aplikasi kendali perangkat elektronik pada *smartphone* android

4.2.3 Prosedur Pengujian

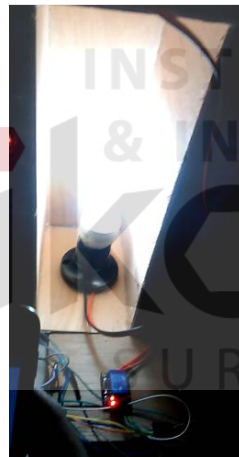
1. Buka Aplikasi kendali perangkat elektronik.
2. Hubungkan *bluetooth smartphone* dengan *bluetooth hc-05*.
3. Dengan aplikasi kendali perangkat elektronik, untuk kontrol lampu terdapat dua tombol yaitu tombol nyala dan mati
4. Untuk kontrol otomatis, dapat dilakukan dengan tombol otomatis.

4.2.4 Hasil Pengujian Kontrol Lampu

Pengujian kontrol lampu dibutuhkan untuk memastikan bahwa komunikasi antara aplikasi dan lampu yang terhubung melalui arduino uno dapat bekerja sesuai keinginan atau tidak. Gambar 4.9 menunjukkan bahwa lampu 1 mati, Gambar 4.10 menunjukkan bahwa lampu 1 menyala, Gambar 4.11 menunjukkan bahwa lampu 2 mati, Gambar 4.12 menunjukkan bahwa lampu 2 menyala, Gambar 4.13 menunjukkan bahwa lampu 3 mati, dan Gambar 4.14 menunjukkan bahwa lampu 3 menyala.



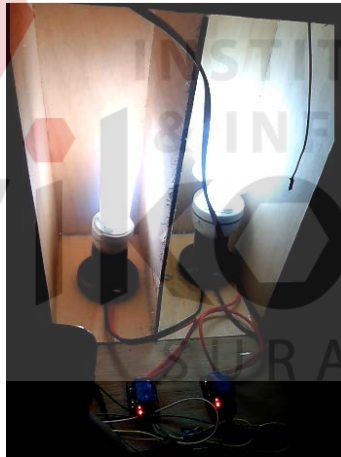
Gambar 4.9 Rangkaian Lampu 1 Dalam Kondisi Mati



Gambar 4.10 Rangkaian Lampu 1 Dalam Kondisi Nyala



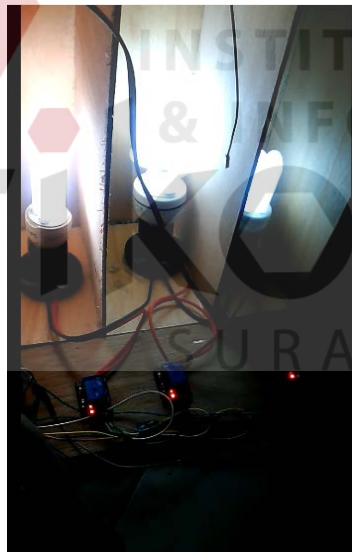
Gambar 4.11 Rangkaian Lampu 2 Dalam Kondisi Mati



Gambar 4.12 Rangkaian Lampu 2 Dalam Kondisi Nyala



Gambar 4.13 Rangkaian Lampu 3 Dalam Kondisi Mati



Gambar 4.14 Rangkaian Lampu 3 Dalam Kondisi Nyala

Untuk tingkat keberhasilan fungsi kontrol lampu perlu dilakukan beberapa kali percobaan yang hasilnya dapat dilihat dari Tabel 4.1.

Tabel 4.1 Hasil Percobaan Kontrol Lampu

Percobaan Ke -	Jenis Tombol	
	Nyala	Mati
1	Berhasil	Berhasil
2	Berhasil	Berhasil
3	Berhasil	Berhasil
4	Berhasil	Berhasil
5	Berhasil	Berhasil
6	Berhasil	Berhasil
7	Berhasil	Berhasil
8	Berhasil	Berhasil
9	Berhasil	Berhasil
10	Berhasil	Berhasil
11	Berhasil	Berhasil
12	Berhasil	Berhasil
13	Berhasil	Berhasil
14	Berhasil	Berhasil
15	Berhasil	Berhasil
16	Berhasil	Berhasil
17	Berhasil	Berhasil
18	Berhasil	Berhasil
19	Berhasil	Berhasil
20	Berhasil	Berhasil
21	Berhasil	Berhasil
22	Berhasil	Berhasil
23	Berhasil	Berhasil
24	Berhasil	Berhasil
25	Berhasil	Berhasil
26	Berhasil	Berhasil
27	Berhasil	Berhasil

Percobaan Ke -	Jenis Tombol	
	Nyala	Mati
28	Berhasil	Berhasil
29	Berhasil	Berhasil
30	Berhasil	Berhasil

Untuk tingkat keberhasilan fungsi alat perlu dilakukan beberapa kali percobaan komunikasi *bluetooth* yang hasilnya dapat dilihat dari Tabel 4.2.

Tabel 4.2 Hasil Percobaan Komunikasi *Bluetooth*

Percobaan Ke -	Jarak Komunikasi	Status
1	1 Meter	Berhasil
2	2 Meter	Berhasil
3	3 Meter	Berhasil
4	4 Meter	Berhasil
5	5 Meter	Berhasil
6	6 Meter	Berhasil
7	7 Meter	Berhasil
8	8 Meter	Berhasil
9	9 Meter	Berhasil
10	10 Meter	Berhasil
11	11 Meter	Berhasil
12	12 Meter	Berhasil
13	13 Meter	Berhasil
14	14 Meter	Berhasil
15	15 Meter	Berhasil
16	16 Meter	Berhasil
17	17 Meter	Berhasil
18	18 Meter	Berhasil
19	19 Meter	Berhasil

Percobaan Ke -	Jarak Komunikasi	Status
20	20 Meter	Berhasil
21	21 Meter	Berhasil
22	22 Meter	Berhasil
23	23 Meter	Berhasil
24	24 Meter	Berhasil
25	25 Meter	Berhasil
26	26 Meter	Berhasil
27	27 Meter	Berhasil
28	28 Meter	Berhasil
29	29 Meter	Berhasil
30	30 Meter	Berhasil

Percobaan komunikasi dilakukan dari berbagai jarak dan dilakukan dengan tidak adanya halangan yang menutupi jalur komunikasi antara modul *bluetooth* dengan *smartphone*.

4.3 Pengujian Pagar

Pengujian pagar dilakukan dengan menguji apakah komunikasi antara aplikasi kendali perangkat elektronik dan rangkaian pagar untuk membuka dan menutup telah berfungsi dengan baik.

4.3.1 Tujuan

Pengujian pagar ini bertujuan untuk menguji apakah motor dc akan bergerak setelah dikontrol melalui aplikasi kendali perangkat elektronik.

4.3.2 Alat yang digunakan

1. Perangkat Keras (Hardware)

- a. *Smartphone* android
 - b. Motor dc 12 V
 - c. *Bluetooth* hc-05
 - d. *Single Board Relay*
2. Perangkat Lunak (*Software*)
 - a. Aplikasi kendali perangkat elektronik pada *smartphone* android.

4.3.3 Prosedur Pengujian

1. Buka Buka Aplikasi kendali perangkat elektronik.
2. Hubungkan *Bluetooth* smartphone dengan *Bluetooth* hc-05.
3. Hubungkan rangkaian pagar dengan arduino dengan kabel USB konverter.
4. Dengan aplikasi kendali perangkat elektronik, akan menampilkan halaman kontrol yang digunakan untuk membuka dan menutup pagar.

4.3.4 Hasil Pengujian Sistem Kontrol Pagar

Pengujian sistem kontrol pagar dibutuhkan untuk memastikan bahwa komunikasi antara aplikasi kendali perangkat elektronik dan motor dc yang terhubung melalui arduino uno dapat bekerja sesuai keinginan atau tidak. Gambar 4.15 menunjukkan bahwa pagar dalam posisi terbuka dan Gambar 4.16 menunjukkan pagar pada posisi menutup setelah tombol tutup dalam aplikasi ditekan.



Gambar 4.15 Pagar pada posisi terbuka



Gambar 4.16 Pagar pada posisi menutup

Untuk tingkat keberhasilan sistem kontrol pagar dalam posisi mengunci pintu perlu dilakukan beberapa kali percobaan yang hasilnya dapat dilihat dari Tabel 4.3.

Tabel 4.3 Hasil Percobaan Sistem Kontrol Pagar

Percobaan Ke -	Jenis Tombol	
	Tutup	Buka
1	Berhasil	Berhasil
2	Berhasil	Berhasil
3	Berhasil	Berhasil
4	Berhasil	Berhasil

Percobaan Ke -	Jenis Tombol	
	Tutup	Buka
5	Berhasil	Berhasil
6	Berhasil	Berhasil
7	Berhasil	Berhasil
8	Berhasil	Berhasil
9	Berhasil	Berhasil
10	Berhasil	Berhasil
11	Berhasil	Berhasil
12	Berhasil	Berhasil
13	Berhasil	Berhasil
14	Berhasil	Berhasil
15	Berhasil	Berhasil
16	Berhasil	Berhasil
17	Berhasil	Berhasil
18	Berhasil	Berhasil
19	Berhasil	Berhasil
20	Berhasil	Berhasil
21	Berhasil	Berhasil
22	Berhasil	Berhasil
23	Berhasil	Berhasil
24	Berhasil	Berhasil
25	Berhasil	Berhasil
26	Berhasil	Berhasil
27	Berhasil	Berhasil
28	Berhasil	Berhasil
29	Berhasil	Berhasil
30	Berhasil	Berhasil

Untuk tingkat keberhasilan fungsi alat perlu dilakukan beberapa kali percobaan komunikasi *bluetooth* yang hasilnya dapat dilihat dari Tabel 4.4.

Tabel 4.4 Hasil Percobaan Komunikasi *Bluetooth*

Percobaan Ke -	Jarak Komunikasi	Status
1	1 Meter	Berhasil
2	2 Meter	Berhasil
3	3 Meter	Berhasil
4	4 Meter	Berhasil
5	5 Meter	Berhasil
6	6 Meter	Berhasil
7	7 Meter	Berhasil
8	8 Meter	Berhasil
9	9 Meter	Berhasil
10	10 Meter	Berhasil
11	11 Meter	Berhasil
12	12 Meter	Berhasil
13	13 Meter	Berhasil
14	14 Meter	Berhasil
15	15 Meter	Berhasil
16	16 Meter	Berhasil
17	17 Meter	Berhasil
18	18 Meter	Berhasil
19	19 Meter	Berhasil
20	20 Meter	Berhasil
21	21 Meter	Berhasil
22	22 Meter	Berhasil
23	23 Meter	Berhasil
24	24 Meter	Berhasil
25	25 Meter	Berhasil

Percobaan Ke -	Jarak Komunikasi	Status
26	26 Meter	Berhasil
27	27 Meter	Berhasil
28	28 Meter	Berhasil
29	29 Meter	Berhasil
30	30 Meter	Berhasil

Percobaan komunikasi dilakukan dari berbagai jarak dan dilakukan dengan tidak adanya halangan yang menutupi jalur komunikasi antara modul *bluetooth* dengan *smartphone*.

