

## **BAB 3**

### **Perancangan Dan Implementasi**

#### **3.1 Analisis sistem**

Kegiatan analisis sistem memegang kunci penting dalam memberikan arahan permasalahan dan menentukan tahap proses pengerjaan selanjutnya dalam hal penentuan kebijakan.

##### **3.1.1 Analisis masalah**

Sistem yang lama menggunakan kendali manual yaitu dengan cara kontak fisik langsung antara pengguna dan saklar lampu sebagai penyambung atau pemutus arus listrik untuk menyala-matikan piranti listrik. Cara manual seperti ini kurang efektif karena bisa terjadi kesalahan manusia seperti lupa dan malas yang dapat membuat penggunaan listrik yang berlebihan. Maka hal tersebut penulis merancang sebuah sistem yang dapat mengendalikan piranti listrik di rumah menggunakan *smartphone android* sebagai media kontrol on/off dari piranti-piranti tersebut.

##### **3.1.2 Analisis kebutuhan**

Analisa kebutuhan dilakukan untuk mengetahui spesifikasi dari kebutuhan aplikasi yang akan dibangun. Pada tahap ini akan membahas mengenai perangkat keras dan perangkat lunak yang digunakan dalam pembuatan prototipe pengendalian piranti listrik berbasis mikrokontroler arduino menggunakan *smartphone android* melalui *WiFi*

###### **3.1.2.1 Kebutuhan perangkat keras**

Adapun perangkat keras yang digunakan untuk membangun perangkat ini adalah sebagai berikut:

- a. Komputer Personal dengan spesifikasi berikut:
  - 1) Processor Ryzen 5 1600 @ 2.80 Ghz
  - 2) Ram 8 Gigabyte
  - 3) *Harddisk* 1 (satu) Terabyte
- b. Laptop Lenovo Ideapad Z480 dengan spesifikasi berikut:
  - 1) Processor Intel Core I3 3110M @ 2.40 Ghz
  - 2) Ram 4 Gigabyte
  - 3) *Harddisk* 1 (satu) Terabyte
- c. *Smartphone Android*
- d. *NodeMcu esp8266*
- e. *Breadboard*
- f. Kabel *jumper*
- g. *Relay Optocoupler 4-way 3,3 Volt Module*
- h. Kabel *USB A to Micro B*
- i. *Voltage Regulator Module LM317*
- j. Adaptor Charger 9 Volt

### **3.1.2.2 Kebutuhan perangkat lunak**

Adapun perangkat lunak yang digunakan untuk membangun perangkat ini adalah sebagai berikut:

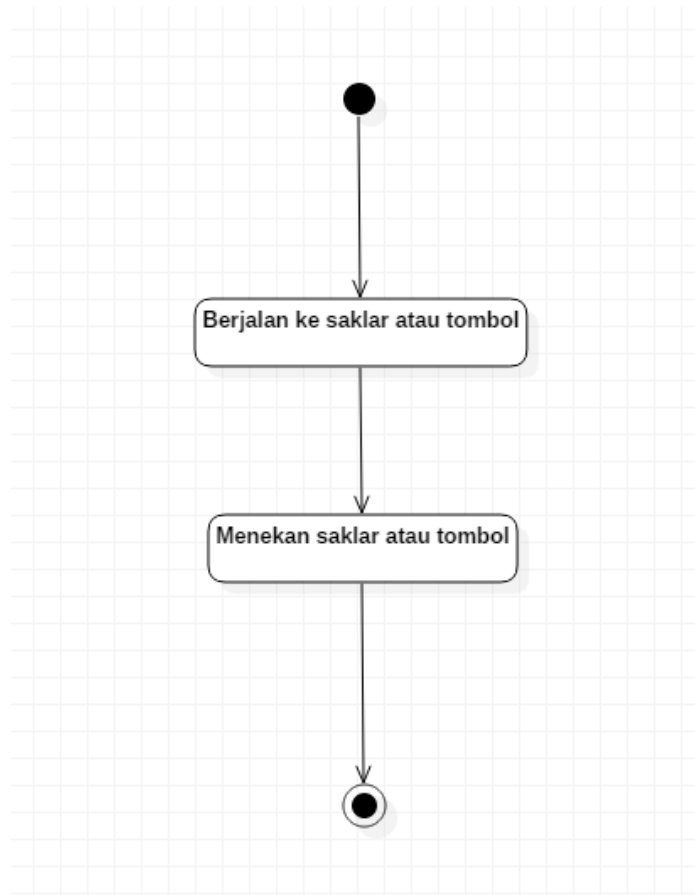
- a. Arduino IDE 1.8.9
- b. Sistem Operasi Windows 10 64bit
- c. *Web Browser Vivaldi*
- d. *Mit App Inventor*

### 3.1.3 Kelayakan teknologi

Secara teknologi, perangkat ini layak digunakan karena teknologi perangkat ini merupakan pengembangan saklar-saklar yang sudah ada dipasaran yang digabungkan dengan *smartphone android* sebagai media kendali melalui teknologi WiFi yang sudah tersedia secara bebas. Perangkat ini juga menggunakan Mikrokontroler *NodeMCU esp8266* sebagai otak pengendalinya dari rangkaian. Teknologi ini cukup inovatif karena penggunaanya dapat dikendalikan dari jarak kurang lebih 10 meter tergantung kondisi lapangan dan juga kekuatan sinyal dari *WiFi Access Point*. Tidak hanya itu saja, pengendalian piranti listrik rumah berbasis mikrokontroler *NodeMCU* menggunakan *smartphone android* ini juga bisa mengendalikan satu atau lebih piranti listrik tergantung kebutuhan.

### 3.1.4 Diagram Menyalakan dan Mematikan alat Listrik

Berikut ini adalah aktivitas menyalakan atau mematikan piranti listrik secara manual.

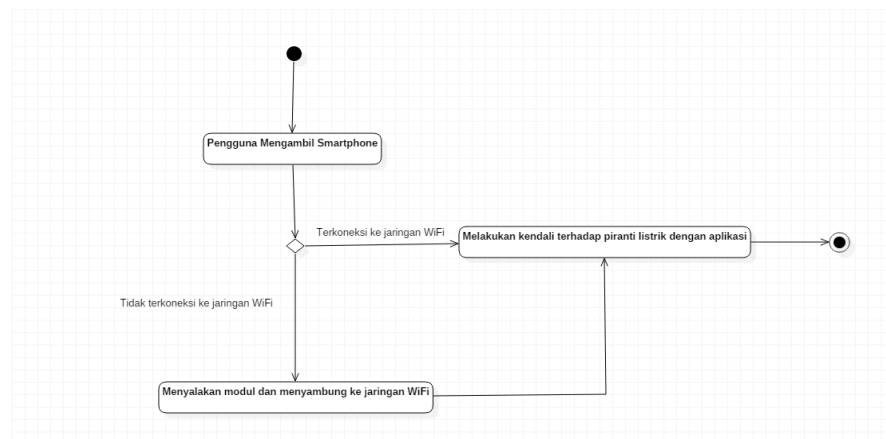


Gambar 3.1 Activity diagram menyalakan atau mematikan piranti listrik

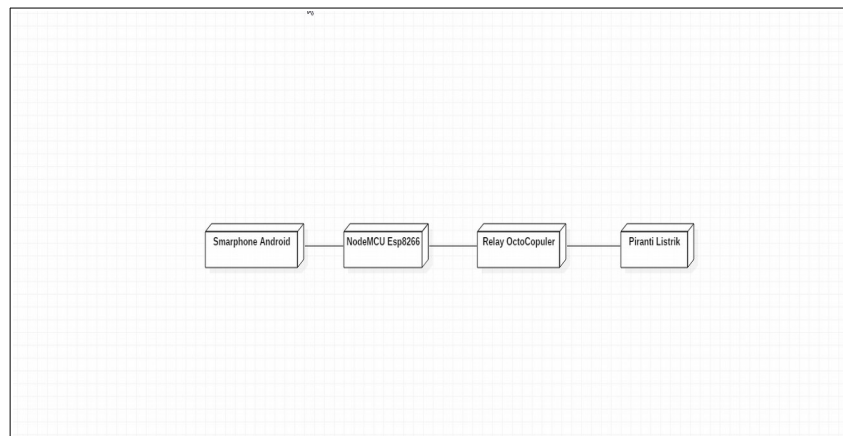
### 3.2 Perancangan Sistem

Dalam sistem yang baru ini, user dapat melakukan input dari aplikasi yang tersedia di perangkat *smartphone android* yaitu dengan cara menekan tombol input (tombol on/off relay). Data yang dikirim adalah sebuah *HTTP GET request* yang dikirim melalui jaringan *WiFi* menggunakan aplikasi yang ada di *smartphone*. Data tersebut akan diterima oleh *NodeMCU esp8266* yang berfungsi

juga sebagai *WiFi Client* dan *web server* yang akan memproses data tersebut, kemudian *NodeMCU esp8266* akan meneruskan data tersebut ke *relay* yang akan menyalakan-matikan piranti listrik yang bersangkutan. Berikut adalah *activity* diagram pada sistem baru



**Gambar 3.2** *activity* diagram dari aliran kerja sistem baru.



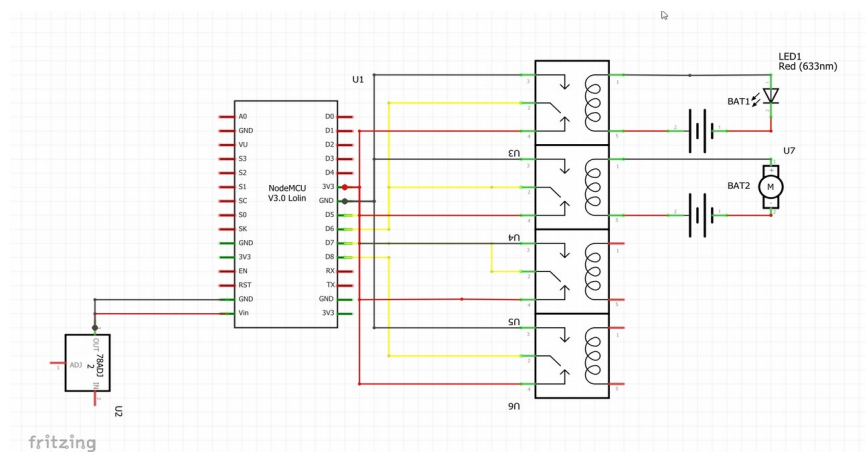
**Gambar 3.3** *Deployment* diagram sistem kendali lampu.

### 3.2.1 Perancangan perangkat keras

Perancangan perangkat keras merupakan rancangan atau rangkaian dari alat yang digunakan untuk membangun prototipe pengendalian piranti listrik berbasis mikrokontroler arduino menggunakan *smartphone android*.

### 3.2.2 Pembuatan rancangan sirkuit

Perancangan sirkuit yang baik dibutuhkan supaya komponen-komponen yang bekerja untuk membuat suatu sistem bekerja dengan baik. Suatu komponen dapat tidak bekerja dengan baik jika terjadi korsleting pada aliran listrik, kekurangan daya yang dapat menyebabkan penurunan kinerja maupun juga kelebihan daya yang melebihi spesifikasi komponen yang dapat memperpendek usia komponen dan membuat panas yang tidak diperlukan.

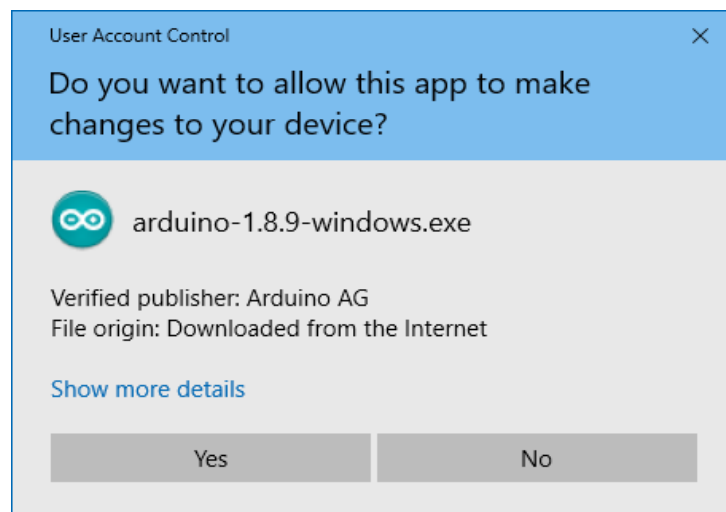


**Gambar 3.4 Rancangan rangkaian sirkuit menggunakan *Fritzing*.**

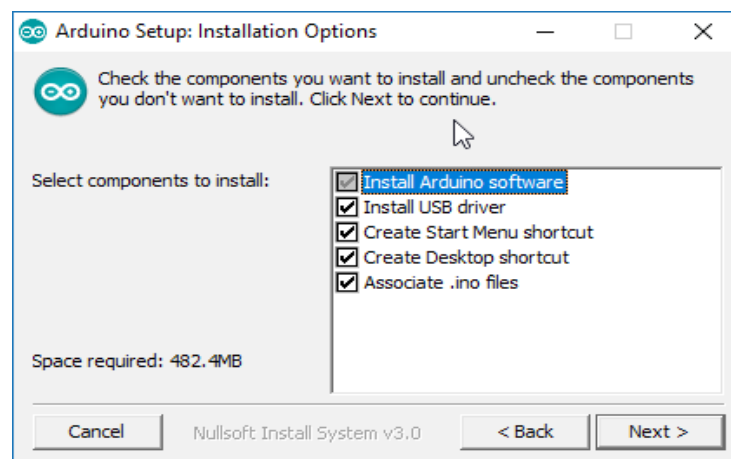
Dengan rangkaian sebagai berikut semua komponen dari mikrokontroler *NodeMCU esp8266*, *relay octocoupler* dan diberi tenaga dari adaptor listrik yang tersambung ke modul LM317 yang akan menurunkan voltase ac 220 volt menjadi voltase dc 3,3 volt yang dapat digunakan oleh *NodeMCU esp8266*.

### 3.3 Instalasi Arduino IDE pada Windows 10

Instalasi Arduino IDE pada sistem komputer berbasis windows dapat dimulai dengan mendownload *software* Arduino IDE versi terbaru di [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software) setelah itu lakukan instalasi dan anda akan konfirmasi klik yes

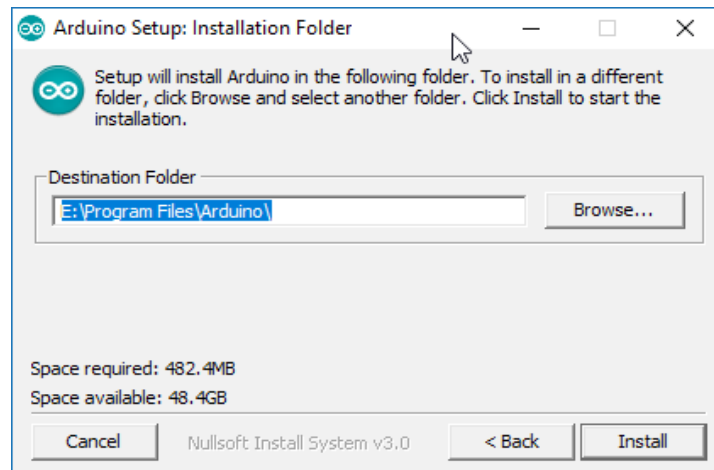


**Gambar 3.5 Konfirmasi User Account Control .**



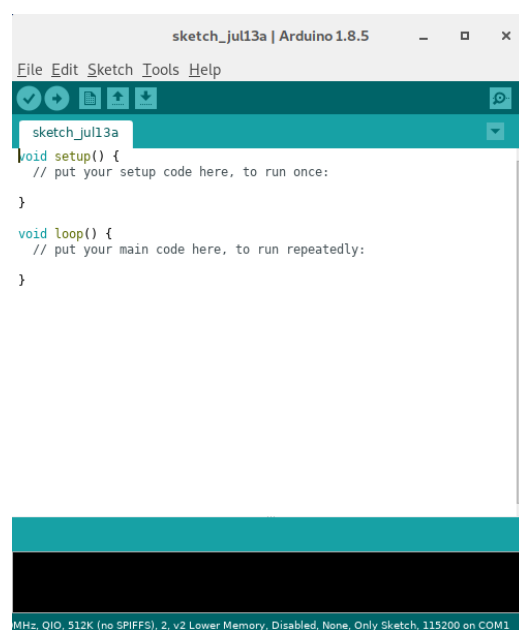
**Gambar 3.6 Opsi instalasi pada Arduino IDE.**

Setelah itu anda dapat memilih modul apa saja yang anda ingin pasang namun untuk pemakaian biasa lanjutkan saja.



**Gambar 3.7 Konfirmasi lokasi instalasi.**

Pilih tempat instalasi sesuai yang anda inginkan atau biarkan seperti setingan semula. Jika semua sudah berjalan dengan lancar maka program akan terpasang di komputer anda dan jika anda membuka aplikasi Arduino IDE maka anda akan disambut dengan layar seperti gambar dibawah ini.

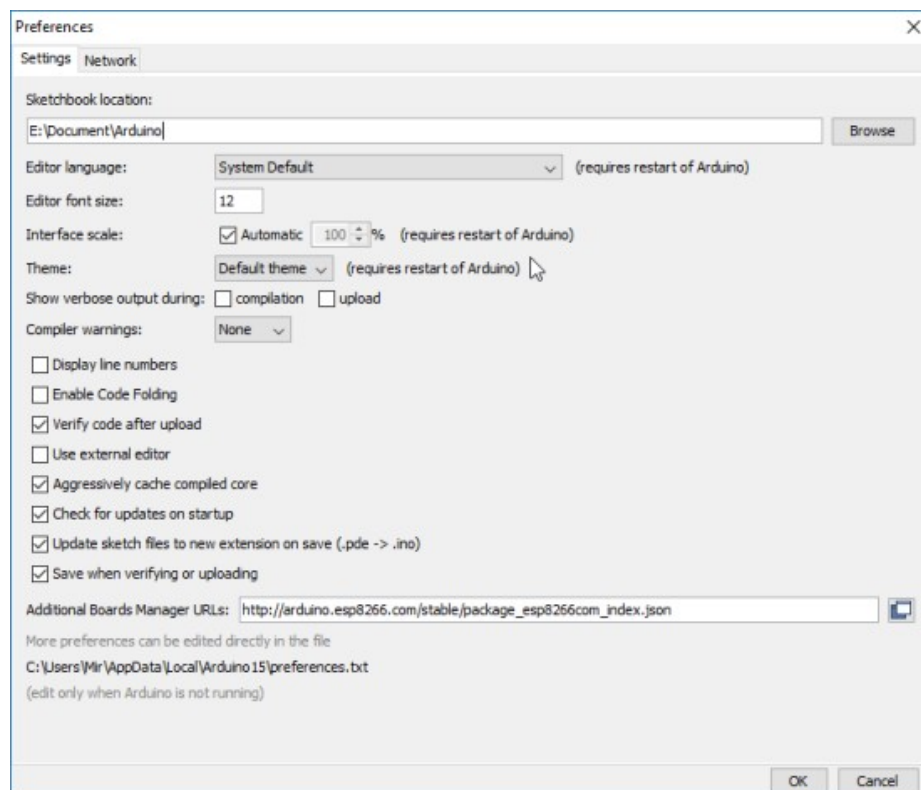


**Gambar 3.7 Tampilan awal Arduino IDE**

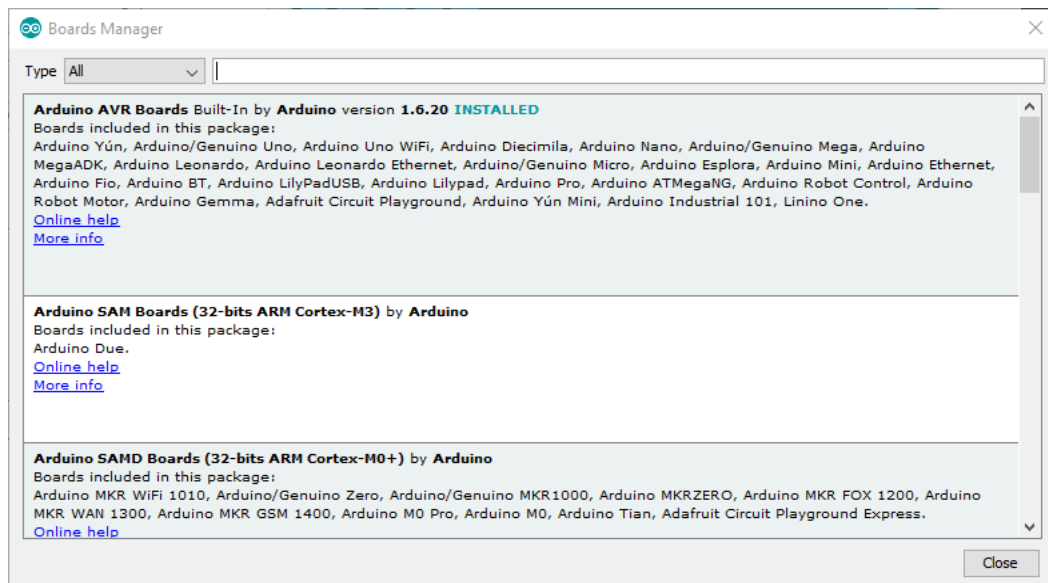


### 3.3.1 Instalasi Library pada Arduino IDE

Untuk dapat menggunakan *NodeMCU esp8266* dengan Arduino IDE maka diperlukan *library* tambahan yang berisi data-data sehingga *NodeMCU esp8266*. Cara untuk menambahkan *library* tersebut dimulai dengan preference masuk kedalam windows preference yang dapat diakses di File > preference, setelah itu pada Additional Board Manager URLs anda harus menambahkan [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) supaya Arduino IDE dapat men-download *library* yang dibutuhkan.

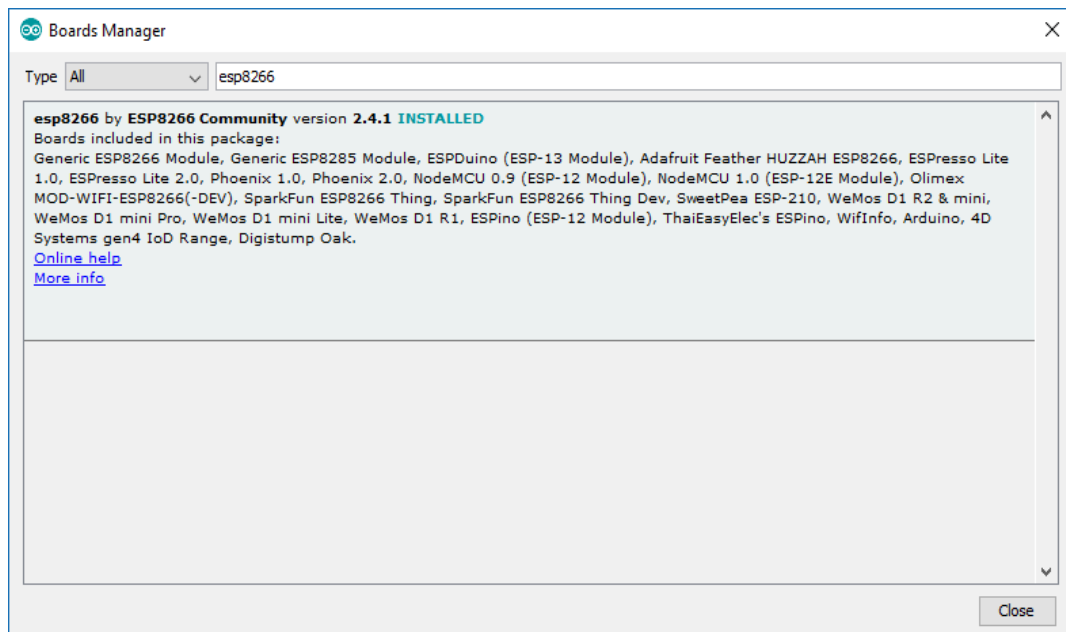


Gambar 3.8 Windows Preference pada Arduino IDE



**Gambar 3.9 Boards manager pada Arduino IDE**

Setelah anda mengikuti langkah sebelumnya anda masih harus mendownload *package library esp8266* di windows Boards Manager yang dapat diakses di Tools > Boards > Boards Manager. Lalu anda dapat menggunakan kotak pencarian dengan mengetikkan *esp8266* lalu klik *download* tunggu sampai selesai. Jika sudah maka anda sudah dapat menggunakan Arduino IDE untuk memprogram *NodeMCU esp8266*.

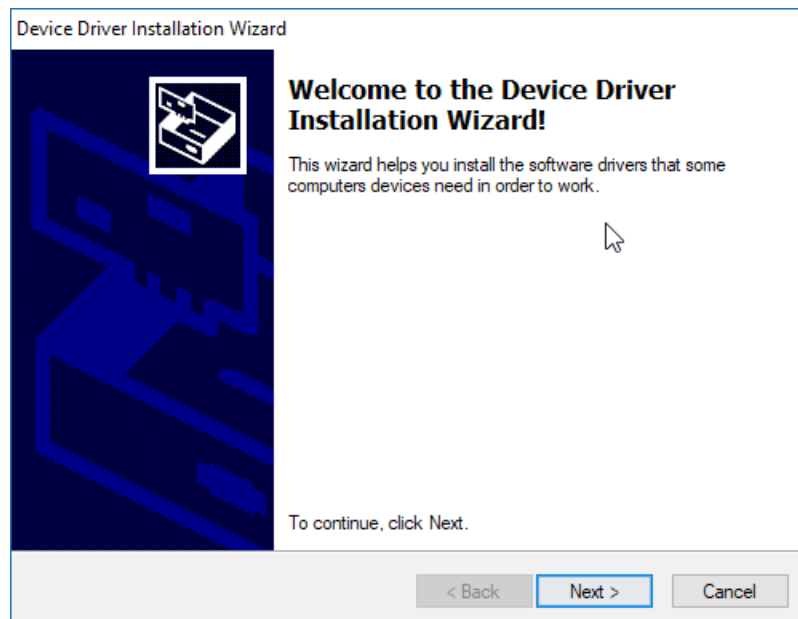


**Gambar 3.10** Hasil pencarian *esp8266* di *Boards Manager*

### 3.3.2 Instalasi *Driver CP2102* pada Windows 10

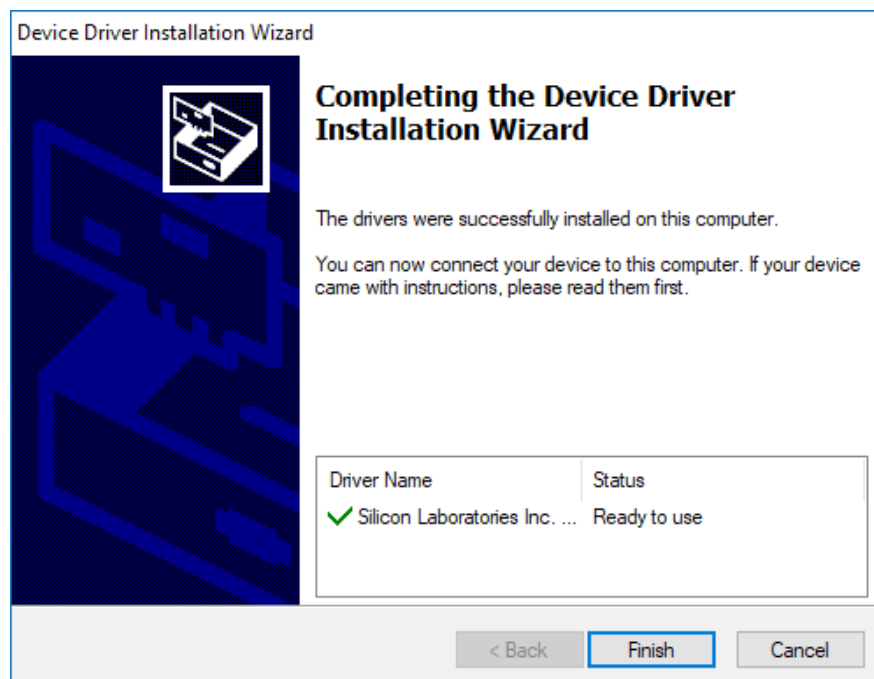
Selain *package library esp8266* dibutuhkan *driver* yang digunakan untuk komputer dapat berkomunikasi dengan *NodeMCU esp8266* melalui *port* USB. Maka dari itu kita perlu memakai *Driver CP2102* yang disediakan oleh silicalabs atau pololu. Untuk mendownload *driver CP2102* dari silicalabs anda harus melakukan pendaftaran akun di website silicalabs yang dapat diakses melalui [www.silabs.com](http://www.silabs.com) setelah mendaftar anda dapat mendownload *Driver* tersebut di websitenya atau mengikuti link ini <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>.

Setelah download selesai anda diharapkan melakukan instalasi *driver* yang sesuai dengan arsitektur dan operating system yang dipakai oleh komputer anda yaitu antara x64 atau x86. Setelah anda memilih dan mengeksekusi *driver* anda akan disambut dengan layar berikut.



Gambar 3.11 Window instalasi driver CP2102

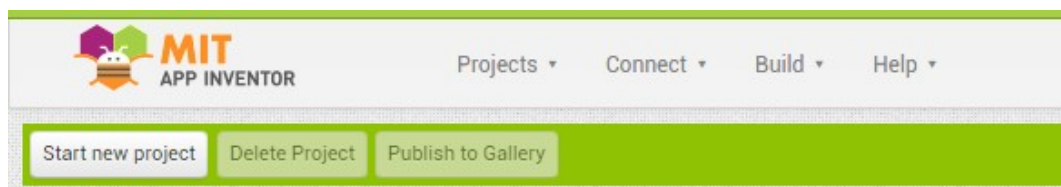
proses pemasangan driver cukup gampang pengguna hanya perlu memencet next dan instalasi selesai



Gambar 3.12 Konfirmasi hasil instalasi

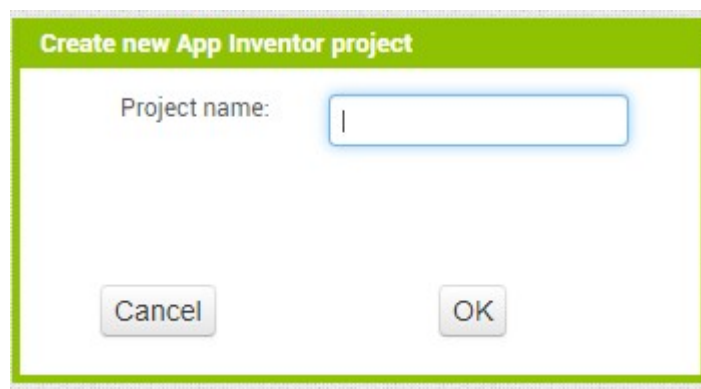
### 3.4 Pembuatan aplikasi *android* menggunakan *MIT App Inventor 2*

Pembuatan aplikasi di program berbasis web *MIT App Inventor 2* dimulai dengan login ke *MIT APP Inventor 2* yang dapat dilakukan dengan cara mengklik tautan <http://ai2.appinventor.mit.edu/> di tautan tersebut pengguna dapat membuat id baru dengan menyambungkan google ataupun dengan facebook sehingga proses tersebut mudah.



**Gambar 3.13 Menu depan dari *MIT App Inventor 2***

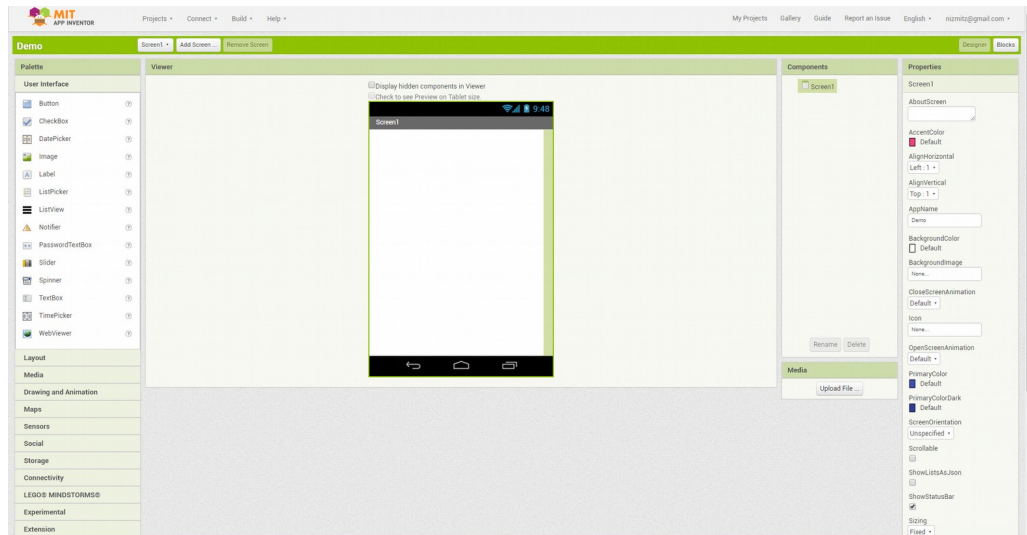
Setelah login pengguna akan disambut dengan halaman depan, di halaman itu juga pembuatan aplikasi dapat dilakukan dengan cara mengklik *start a new project* dan sebuah jendela baru akan terbuka untuk memberi nama pada proyek yang akan dibuat.



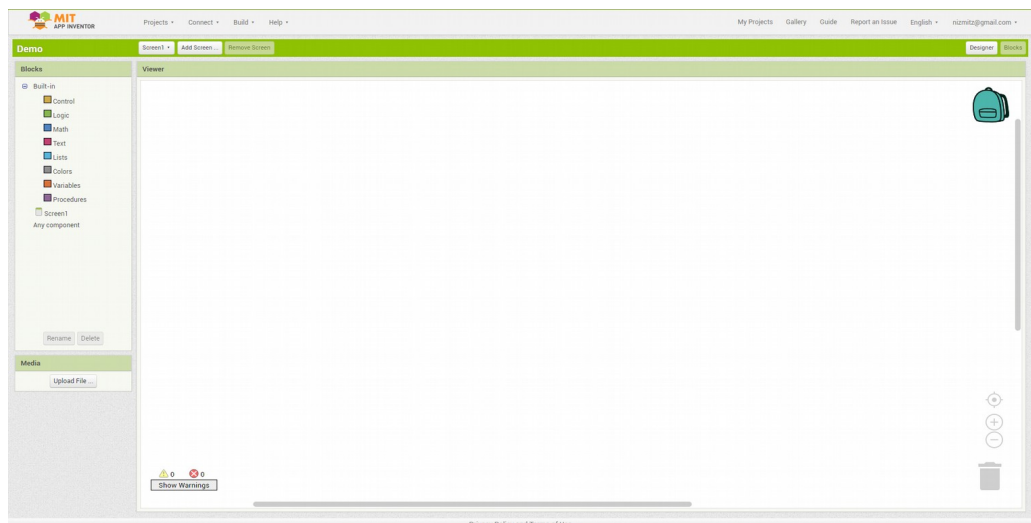
**Gambar 3.14 Jendela untuk memberi nama proyek**

Tampilan berikut adalah tampilan kerja yang dapat digunakan untuk membuat program terdapat dua buah tampilan kerja yaitu *designer* dan *block* yang berfungsi berbeda yaitu *designer* untuk membuat tampilan pengguna dapat dilihat

di gambar dan *block* digunakan untuk membuat logika dari program yang dapat dilihat di gambar.



**Gambar 3.15 Tampilan *designer* dari MIT App Inventor 2**

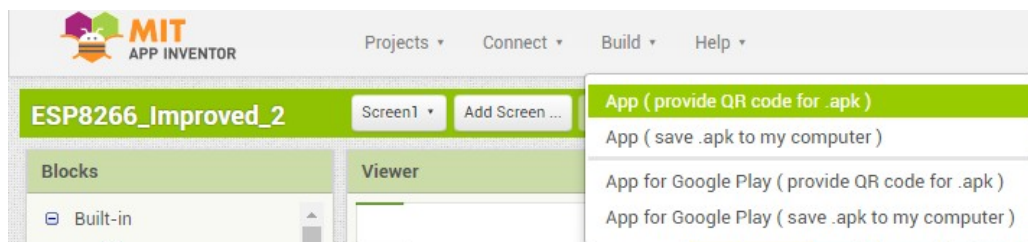


**Gambar 3.16 Tampilan block dari MIT App Inventor 2**

### 3.5 Membuat file APK

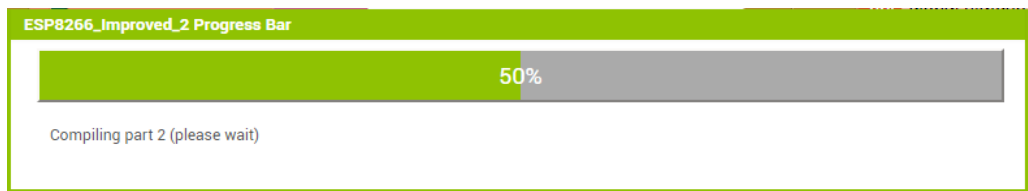
Pada MIT App Inventor 2 terdapat dua cara untuk membuat aplikasi yang kita buat menjadi sebuah file APK yang dapat di instalasikan ke *smartphone*

*android* yaitu dengan kita mendownload APKnya ke komputer lalu kita transfer APK tersebut ke *smartphone android* melalui *bluetooth* maupun kabel *usb*. Atau menggunakan cara kedua dengan cara membuat apk di web lalu memberikan tautan dengan QR code yang dapat dipindai oleh kamera *smartphone android*, lalu *smartphone android* akan mengunduh data aplikasi langsung dari website MIT App Inventor 2.



**Gambar 3.14 Pilihan untuk membuat APK**

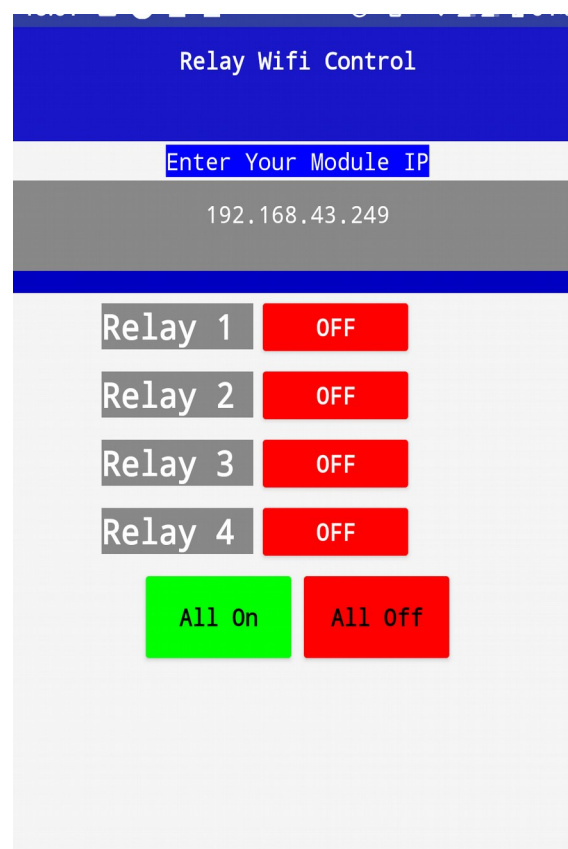
Setelah memilih salah satu dari pilihan kita akan disambut dengan *pop up* proses kemajuan dari pembuatan APK seperti di gambar berikut.



**Gambar 3.15 Progress bar pembuatan APK pada MIT App Inventor 2**

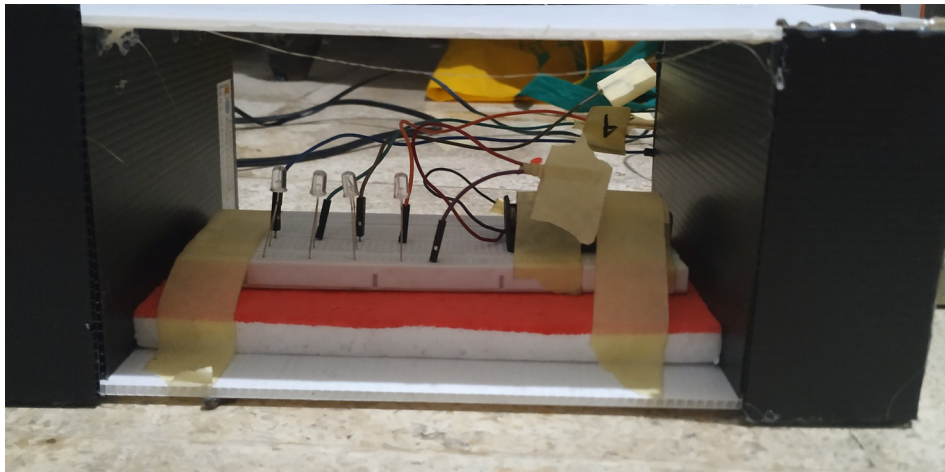
### 3.6 Uji coba aplikasi

Tahap ini merupakan tahapan uji coba pada perangkat *android* setelah instalasi dilakukan menggunakan cara-cara yang terdapat diatas. Gambar-gambar dibawah merupakan gambar dari tatapan antarmuka dari aplikasi yang dibuat menggunakan MIT App Inventor 2.

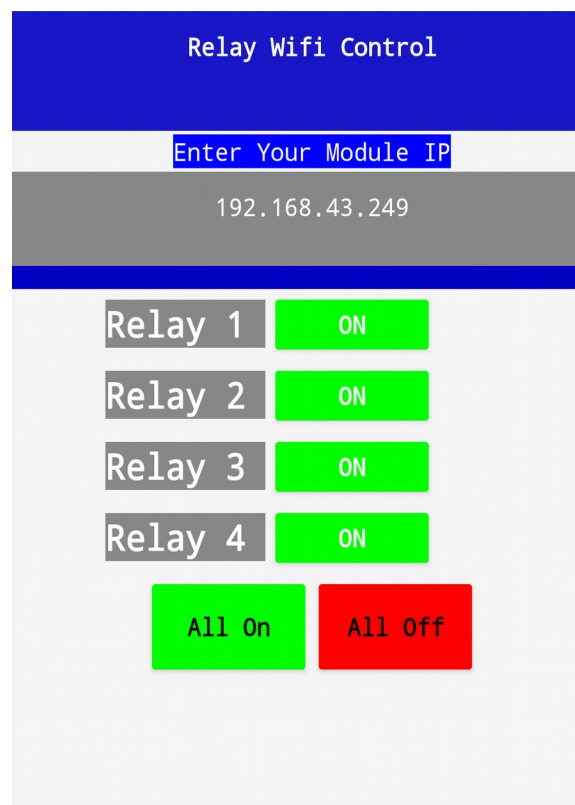


Gambar 3.19 Tampilan aplikasi dengan kondisi semua *button relay off*

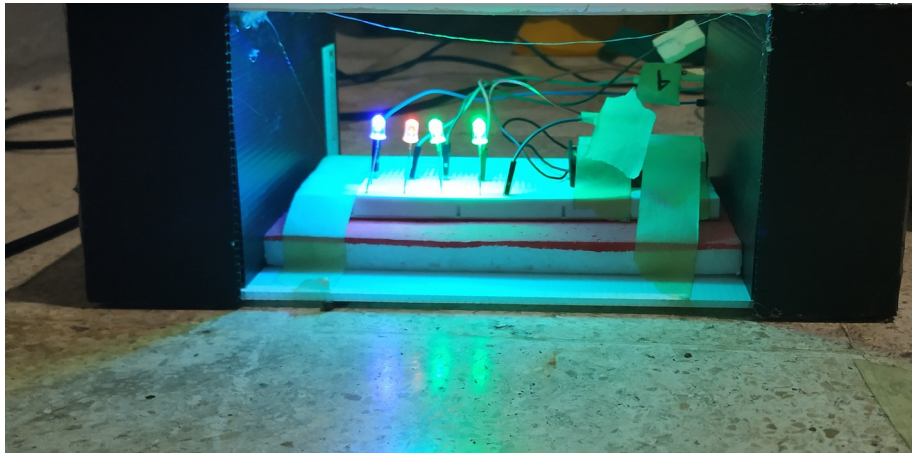




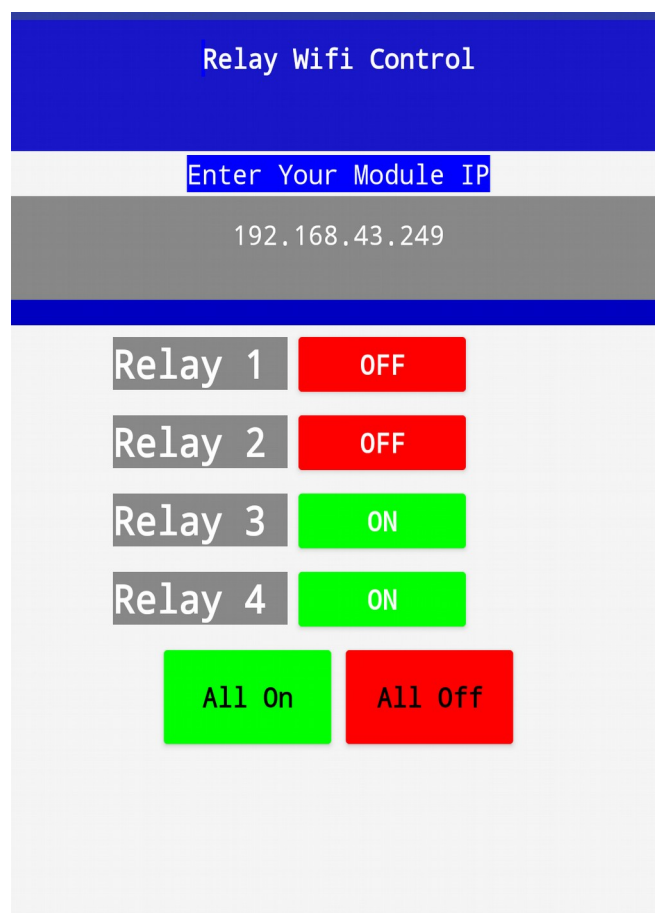
Gambar 3.20 Foto percobaan dengan kondisi semua *button relay off*



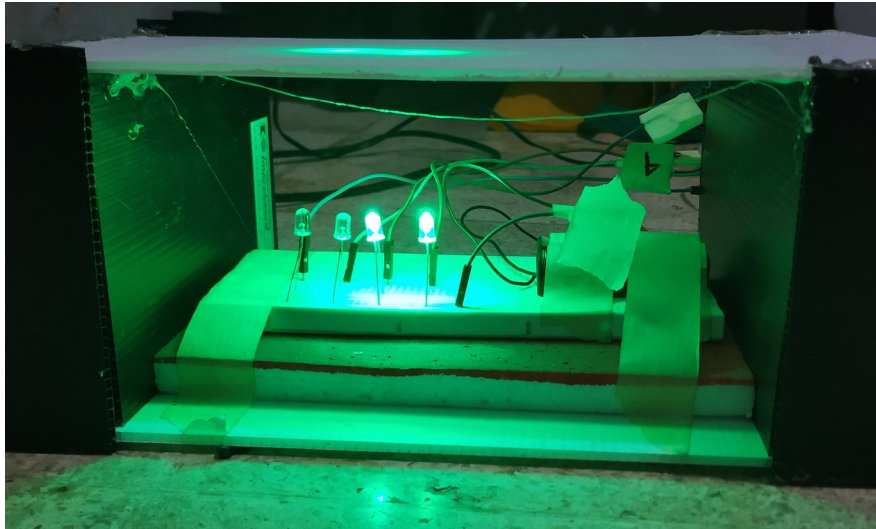
Gambar 3.21 Tampilan aplikasi dengan semua *button relay on*



Gambar 3.22 Foto percobaan dengan kondisi semua *button relay on*



Gambar 3.23 Tampilan aplikasi dengan 2 buah *button relay on*.



**Gambar 3.24** Foto percobaan dengan kondisi 2 buah *button relay on*.

Aplikasi pengendalian piranti listrik ini dapat berfungsi menggunakan dua jenis mode yang dapat disesuaikan dengan ada tidaknya *WiFi Access Point*. Jika tersedia *WiFi router* yang dapat digunakan oleh modul wifi dari NodeMCU esp8266 maka modul tersebut akan tersambung dengan *WiFi* yang tersedia. Namun jika tidak terdapat *WiFi router* atau piranti lain yang dapat membuat *Access Point* maka modul Wifi NodeMCU esp8266 juga dapat berfungsi sebagai *WiFi Access Point* dan piranti *android* dapat menyambung ke *WiFi* tersebut untuk melakukan pengendalian piranti listrik

### 3.6.2 Uji coba aplikasi *android* Xiaomi Mi A2 sebagai *Access Point*

Tabel 3.1 observasi relay dengan Xiaomi Mi A2 sebagai *Access Point*

Jarak (Meter) dan kondisi	Sinyal Wifi	Nomor Relay	Hasil Observasi
1 Meter, Jarak pandang mata	<i>Connected</i>	<i>Relay 1</i> <i>Relay 2</i> <i>Relay 3</i> <i>Relay 4</i>	Menyala setelah ditekan on, dan mati ketika ditekan tombol off. Ketika tombol All On/All Off semua <i>relay</i> mati/nyala. Keterlambatan hampir tidak dirasakan

Tabel 3.1 observasi relay dengan Xiaomi Mi A2 sebagai Access Point (Lanjutan)

Jarak (Meter) dan kondisi	Sinyal Wifi	Nomor Relay	Hasil Observasi
3 Meter, Jarak pandang mata	<i>Connected</i>	<i>Relay 1</i> <i>Relay 2</i> <i>Relay 3</i> <i>Relay 4</i>	Menyala setelah ditekan on, dan mati ketika ditekan tombol off. Ketika tombol All On/All Off semua <i>relay</i> mati/nyala. Keterlambatan tidak dirasakan
8 Meter, Jarak pandang mata	<i>Connected</i>	<i>Relay 1</i> <i>Relay 2</i> <i>Relay 3</i> <i>Relay 4</i>	Menyala setelah ditekan on, dan mati ketika ditekan tombol off. Ketika tombol All On/All Off semua <i>relay</i> mati/nyala. Keterlambatan tidak dirasakan
10 Meter, Tidak dalam jarak pandang mata, tertutupi dinding	<i>Not Connected</i>	<i>Relay 1</i> <i>Relay 2</i> <i>Relay 3</i> <i>Relay 4</i>	Tombol <i>responsive</i> namun tidak terjadi perubahan apapun pada <i>relay</i> .

Hal perlu diperhatikan bahwa tabel diatas dicoba dengan kondisi yang tidak dikondisikan dengan arti lain akan terjadi interferensi antara *WiFi Access Point*, dan sumber lain yang memancar gelombang mikrowave pada band 2,4ghz.

Tabel 3.2 Kompatibilitas aplikasi pada *smartphone*

Tipe ponsel pintar	Spesifikasi	Versi Sistem Operasi	Hasil
Xperia Z2	Snapdragon 801, Ram 3GB, Rom 32GB, Wi-Fi ac/b/g/n	6.0 Marshmallow, Xperia Launcher	UI Sesuai dengan rancangan
Xperia SP	Snapdragon S4 Pro, Ram 1GB, Rom 8GB, Wi-Fi b/g/n	4.3 Jelly Bean, Xperia Launcher	UI Sesuai dengan rancangan
Mi A2	Snapdragon 660 Ram 4, Rom 64, Wi-Fi ac/b/g/n	Android 9 (Pie) Google launcher	UI Sesuai dengan rancangan

Aplikasi tetap berjalan lancar di *android* versi lama maupun *android* versi terkini. Selain itu aplikasi juga tetap dapat menjaga antarmuka walaupun disajikan pada *smartphone* yang memiliki aspect ratio, resolusi, dan ukuran layar yang berbeda.