BAB I

PENDAHULUAN

A. Latar Belakang

Internet merupakan dunia baru yang penuh pesona saat pertama kali muncul dan pertama kali dikenalkan, internet terus memikat untuk di ekplorasi lebih lanjut, digali dan di kembangkan oleh para ahli teknologi. para ahli teknologi. menciptakan inovasi-inovasi untuk memudahkan pekerjaan demi kinerja yang lebih baik. Munculnya sistem operasi *Android* yang dianggap sebagian besar manusia adalah sistem operasi yang mudah dioperasikan, *fleksibel*, dan tidak terbatas untuk membangun aplikasi sendiri, menyebabkan menjamurnya aplikasi buatan pengguna itu sendiri

Seiring dengan semakin berkembangnya infrastruktur internet, bahkan bukan hanya *smartphone* dan komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Sebagai contoh dapat berupa peralatan elektronik, peralatan yang digunakan manusia, dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Di dunia bidang *"IT"*, konsep ini telah dikenal dengan istilah *"Internet of Things"* atau yang disebut dengan singkatan *IoT*. *Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. *Internet of Things (IoT)* mengacu pada benda yang diidentifikasi secara unik sebagai reservasi virtual dalam struktur berbasis

internet. *Internet of Things (IoT)* sebagai sebuah infrastruktur jaringan global, yang menghubungkan benda-benda fisik dan virtual melalui eksploitasi data capture dan kemampuan komunikasi dengan sensor dan koneksi sebagai pengembangan layanan. Dalam hal tersebut dapat disimpulkan bahwa *IoT* mengacu dan memanfaatkan pada suatu benda yang nantinya benda tersebut akan dapat berkomunikasi antara satu dengan yang lain melalui sebuah jaringan internet. Salah satu dari penerapan *IoT* adalah pada kendali lampu.

Sistem penerangan yang ada di lingkungan sekolah sebagian besar masih menggunakan cara manual, yang mana untuk menyalakan dan mematikan lampu harus dilakukan secara tradisonal atau dengan bantuan manusia. Hal ini sangat menyulitkan karena mengharuskan guru maupun pegawai secara bergantian datang ke sekolah pada sore hari hanya untuk menyalakan lampu dengan menekan saklar yang ada di sekolah. Kelebihan penelitian ini dari penelitian sebelumnya yaitu prngontrolan sistem lampu dapat dilakukan menggunakan perintah suara dengan mengucapkan format perintah yang telah diatur sebelumnya. Selain itu juga, pengguna dapat menentukan waktu kapan lampu menyala dan kapan lampu akan mati selama terkoneksi dengan internet.

Berdasarkan uraian tersebut di atas, maka pada tugas akhir ini akan dibuat Sistem Kontrol Lampu Berbasis *Internet of Things* dan *Android* pada SDN 2 Lawawoi. Diharapkan aplikasi ini dapat menbantu dalam hal memberi kenyamanan dan kemudahan bagi para peggunan dalam aktivitas sehari – hari.

B. Rumusan masalah

Rumusan masalah dalam proposal penelitian ini adalah Bagaimana Merancang Sistem Kontrol Lampu Berbasis *Internet of Things* dan *Android* pada SDN 2 Lawawoi.

C. Batasan Masalah

Agar pembahasan tidak menyimpang dari tujuan semula, maka penulis membatasi masalah ini hanya pada Aplikasi khususnya yang berkenaan tentang sistem kontrol lampu berbasis *Internet of Things* dan *Android* pada SDN 2 Lawawoi.

D. Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut :

1. Bagi peneliti Dapat memberikan masukan untuk pengembangan ilmu serta referensi untuk peneliti selanjutnya.
2. Bagi SDN 2 Lawawoi dapat di terapkan sistem *control* lampu berbasis *internet of things* dan *android* agar bisa membantu aktivitas.
3. Dalam dunia pendidikan di anggap mampu memberi wawasan mengenai penerapan sistem *control* lampu *internet of things* dan *android*.

E. Tujuan Penelitian

Tujuan dari Penelitian ini adalah untuk menghasilkan suatu aplikasi yang dapat mengendalikan lampu penerangan pada SDN 2 Lawawoi dengan

menggunakan *internet of things dan smartphone android* sehingga memberi kenyamana dan kemudahan dalam kehidupan sehari-hari.

F. Sistematika Penulisan

Untuk mempermudah penulisan dan pembahasan selanjutnya, maka uraian pembahasan skripsi ini ditulis secara sistematis sebagai berikut :

BAB I : **PENDAHULUAN**, membahas tentang latar belakang masalah pokok permasalahan, batasan masalah, tujuan dan manfaat penelitian dan sistematika penulisan.

BAB II : **TINJAUAN PUSTAKA**, membahas tentang kerangka fikir dan landasan teori yang berhubungan dengan aplikasi perangkat lunak pengolahan limbah plastik.

BAB III : **METODOLOGI PENELITIAN**, membahas tentang metode penelitian dan alat yang digunakan untuk mendesain aplikasi.

BAB IV : **HASIL DAN PEMBAHASAN**, membahas tentang analisis sistem, rancangan sistem, implementasi, serta pengujian *prototype.*

BAB V : **PENUTUP**, membahas tentang kesimpulan dan saran-saran yang diperoleh dari seluruh pembahasan.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terkait

Dalam penyusunan skripsi ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini. Berikut ini penelitian yang berhubungan antara lain :

Penelitian yang dilakukan oleh Andri Susanto & Ismail Darisman Jauhari, 2018, "Rancang Bangun Aplikasi *Android* Untuk Kontrol Lampu Gedung Menggunakan Media *Bluetooth* Berbasis *Arduino Uno* dengan metode *BlackBox*"[1]. Ada Empat Tahapan yang digunakan dalam penelitian ini yakni (1) Pengumpulan Data dan Informasi, (2) Analisa Kebutuhan, (3) Desain dan Perancangan Sistem, (4) Pembuatan Aplikasi, (5) Implementasi dan Testing. Dengan Hasil menunjukkan bahwa Aplikasi pada *Android* dapat berkomunikasi dengan *arduino* dengan jarak <12 meter. Aplikasi yang dibuat dapat melakukan 3 perintah yaitu dengan perintah tombol(*ON/OFF*), perintah suara (*speech recognition*), dan perintah *countdown timer*. Alat ini telah teruji dan bisa dijadikan media *android smartphone light controller* via koneksi *Bluetooth* tanpa harus menekan tombol lampu.

Penelitian yang dilakukan oleh Leo Ade Putra & Arif Rahman Hakim, 2018, "Sistem Kendali Lampu Cerdas Pada *Smarthome* Berbasis Android mengunakan Metode *Fuzzy Logic Control*"[2]. Ada Tujuh Tahapan yang

digunakan dalam penelitian ini yakni (1) Studi Pendahuluan, (2) Perumusan Masalah, (3) Menentukan Tujuan Penelitian, (4) Studi Literatur, (5) Pengembangan Desain Sistem, (6) Perancangan Produk, (7) Pengujian Produk. Smartphone yang menggunakan sistem *Android* bisa mengontrol lampu misalnya menyalakan, mematikan, serta mengatur tingkat kecerahan lampu secara manual dan otomatis dengan menggunakan koneksi *Wi-Fi*. Menurut Leo Ade Putra & Arif Rahman Hakim (2018) hasil pengujian proses *defuzzifikasi, fuzzy logic control* tipe Sugeno dapat diterapkan dengan baik pada peralatan rumah tangga khususnya lampu. Hasilnya sama dengan pengujian pada program *Matlab*, Arduino dan dihitung secara manual yang menghasilkan nilai *output* yang sama. Dengan menggunakan *fuzzy*, lampu bisa menyesuaikan kecerahan berdasarkan kondisi pencahayaan di dalam dan di luar ruangan.

Andrianto & Arief Susanto, 2015, "Aplikasi Pengontrol Jarak Jauh Pada Lampu Rumah Berbasis *Android*"[3]. Ada Tiga Tahapan yang digunakan dalam penelitian ini yakni (1) Metode Pengumpulan Data, (2) Metode Analisa, (3) Metode Perancangan. dari pengujian yang dikendalikan melalui *web server* yang dapat mengendalikan serta menerima status dari lampu yang dikontrol menunjukan bahwa pengendalian berjalan sangat baik. Dapat dilihat bahwa semua *GPIO* yang telah diaktifkan mampu mengirimkan perintan untuk menyalakan ataupun mematikan lampu.

B. Pengertian *Speech Recognition*

Speech recognition (SR)—adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasikan kata-kata tersebut. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi sebagai sebuah komando untuk melakukan suatu pekerjaan, misalnya penekanan tombol pada telepon genggam yang dilakukan secara otomatis dengan komando suara.

Alat pengenal ucapan, yang sering disebut dengan *speech recognizer*, membutuhkan sampel kata sebenarnya yang diucapkan dari pengguna. Sampel kata akan didigitalisasi, disimpan dalam komputer, dan kemudian digunakan sebagai basis data dalam mencocokkan kata yang diucapkan selanjutnya. Sebagian besar alat pengenal ucapan sifatnya masih tergantung kepada pembicara. Alat ini hanya dapat mengenal kata yang diucapkan dari satu atau dua orang saja dan hanya bisa mengenal kata-kata terpisah, yaitu kata-kata yang dalam penyampaiannya terdapat jeda antar kata.

C. Konsep Dasar Sistem

1. Pengertian Sistem

Menurut Bambang (2010:12). **S**istem adalah prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan maksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.

Menurut Hardiansyah (2009:10). Bahwa sistem adalah sebuah struktur konseptual yang tersusun dari fungsi-fungsi yang saling berhubungan yang bekerja sebagai suatu kesatuan organik untuk mencapai suatu hasil yang diinginkan secara efektif dan efesien.Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya.

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya didefinisikan oleh Indrajani (2011:12) sebagai setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat – sifat yang tertentu yaitu:

a) Komponen Sistem

Suatu sistem terdiri menurut sejumlah komponen yang saling berinteraksi atau saling berhubungan buat menciptakan suatu kesatuan. Komponen –

komponen sistem atau elemen – elemen sistem dapat berupa suatu subsistem atau bagian – bagian dari sistem.

b) Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

c) Lingkungan Luar Sistem

Lingkungan luar sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem yang bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.

d) Penghubung Sistem

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung.

e) Masukan Sistem

Masukan *(input*) adalah energi yang dimasukkan kedalam sistem, dapat berupa masukan perawatan (*Maintenance Input*) dan masukan sinyal (*Signal Input*).

f) Keluaran Sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan di klasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

Keluaran dapat merupakan masukan untuk sub sistem yang lain atau kepada supra sistem.

g) Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

h) Sasaran Sistem

Menurut Kadir Abdul (2015) Suatu sistem mempunyai tujuan atau sasaran, sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

D. *Internet of Things (IoT)*

Internet of Things (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. *IoT* telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems (MEMS)*, dan Internet.

"A Things" pada Internet of Things dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan *transponder biochip*, sebuah mobil yang telah dilengkapi *built-in sensor* untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, *IoT* paling erat hubungannya dengan komunikasi *machine-to-machine (M2M)* di bidang manufaktur dan listrik, perminyakkan, dan gas. Produk dibangun dengan

kemampuan komunikasi *M2M* yang sering disebut dengan sistem cerdas atau "*smart*". Sebagai contoh yaitu *smart* kabel, *smart meter, smart grid sensor*.

Penelitian pada *IoT* masih dalam tahap perkembangan. Oleh karena itu, tidak ada definisi dari *Internet of Things*. Berikut adalah beberapa definisi alternatif dikemukakan untuk memahami *Internet of Things (IoT)*, antara lain (id.wikipedia.org):

Menurut Ashton pada tahun 2009 definisi awal *IoT* adalah *Internet of Things* memiliki potensi untuk mengubah dunia seperti pernah dilakukan oleh Internet, bahkan mungkin lebih baik. Pernyataan tersebut diambil dari artikel sebagai berikut:

"Hari ini komputer dan manusia, hampir sepenuhnya tergantung pada Internet untuk segala informasi yang semua terdiri dari sekitar 50 *petabyte* (satu *petabyte* adalah 1.024 *terabyte*) data yang tersedia pada Internet dan pertama kali digagas dan diciptakan oleh manusia. Dari mulai magnetik, menakan tombol rekam, mengambil gambar digital atau memadai kode bar.

Diagram konvensional dari Internet meninggalkan *router* menjadi bagian terpenting dari semuanya. Masalahnya adalah orang memiliki waktu, perhatian dan akurasi terbatas. Mereka semua berarti tidak sangat baik dalam menangkap berbagai data tentang hal di dunia nyata.

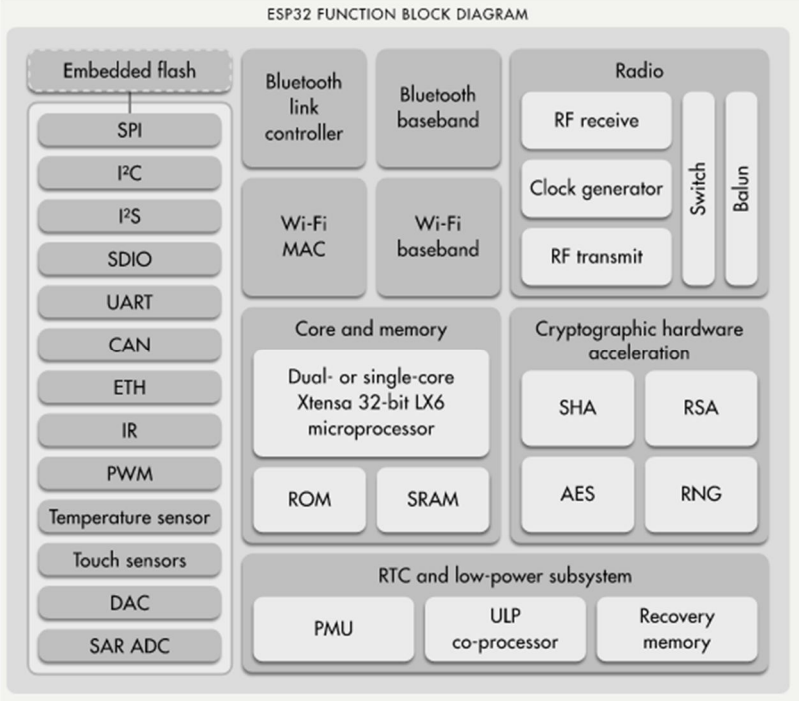
Dari segi fisik dan begitu juga lingkungan kita. Gagasan dan informasi begitu penting, tetapi banyak lagi hal yang pernting. Namun teknologi informasi saat ini sangat tergantung pada data yang berasal dari orang-orang sehingga komputer kita tahu lebih banyak tentang semua ide dari hal-hal tersebut"

E. Esp32

ESP32 dibuat oleh *Espressif Systems, ESP32* adalah sistem dengan biaya yang rendah, berdaya rendah pada seri *chip (SoC)* dengan *Wi-Fi* & kemampuan *Bluetooth* dua mode! Keluarga *ESP32* termasuk chip *ESP32-D0WDQ6 (dan ESP32-D0WD), ESP32-D2WD, ESP32-S0WD,* dan sistem dalam paket (SiP) *ESP32-PICO-D4*. Pada intinya, ada *mikroprosesor Tensilica Xtensa LX6 dual-core* atau *single-core* dengan *clock rate* hingga *240 MHz. ESP32* sudah terintegrasi dengan *built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules.* Didesain untuk perangkat seluler, perangkat elektronik yang dapat dipakai, dan aplikasi *IoT, ESP32* juga bekerja dengan konsumsi daya sangat rendah melalui fitur hemat daya termasuk *fine resolution clock gating, multiple power modes, and dynamic power scaling.*

Module ESP32 merupakan penerus dari module *ESP8266* yang cukup populer untuk Aplikasi *IoT*. Pada *ESP32* terdapat inti *CPU* serta *Wi-Fi* yang lebih cepat, *GPIO* yang lebih, dan mendukung *Bluetooth Low Energy.*

Pada *mikrokontroler* ini sudah tersedia modul *WiFi* dalam *chip* sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. terlihat pada gambar di atas merupakan *pin out* dari *ESP32. Pin* tersebut dapat dijadikan *input* atau *output* untuk menyalakan *LCD,* lampu, bahkan untuk menggerakan motor *DC.*



Gambar 2.1 Arsitektur dan *Block* Diagram dari *ESP32*
(Sumber: Edukasi Elektronika)

Fitur dan Spesifikasi *ESP32*:

1. *Processors*
   a) *Main processor: Tensilica Xtensa 32-bit LX6 microprocessor*
   b) *Cores: 2 or 1 (depending on variation)*
   c) *Clock frequency: up to 240 MHz*
   d) *Performance: up to 600 DMIPS*
   e) *Ultra low power co-processor: allows you to do ADC conversions, computation, and level thresholds while in deep sleep.*
2. *Wireless connectivity*
   a) *Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz up to 150 Mbit/s)*
   b) *Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)*

3. *Memory:*

   a) *Internal memory*

      ROM: 448 KiB

      SRAM: 520 KiB

      RTC fast SRAM: 8 KiB

      RTC slow SRAM: 8 KiB

      eFuse: 1 Kibit

   b) *Embedded flash:*

      0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips)

      2 MiB (ESP32-D2WD chip)

      4 MiB (ESP32-PICO-D4 SiP module)

   c) *External flash & SRAM: ESP32 supports up to four 16 MiB external QSPI flashes and SRAMs with hardware encryption based on AES to protect developers' programs and data. ESP32 can access the external QSPI flash and SRAM through high-speed caches.*

      1) *Up to 16 MiB of external flash are memory-mapped onto the CPU code space, supporting 8-bit, 16-bit and 32-bit access. Code execution is supported.*

      2) *Up to 8 MiB of external flash/SRAM memory are mapped onto the CPU data space, supporting 8-bit, 16-bit and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.*

4. *Peripheral input/output*

   Rich peripheral interface with DMA that includes capacitive touch, ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I²C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I²S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.

5. *Security*

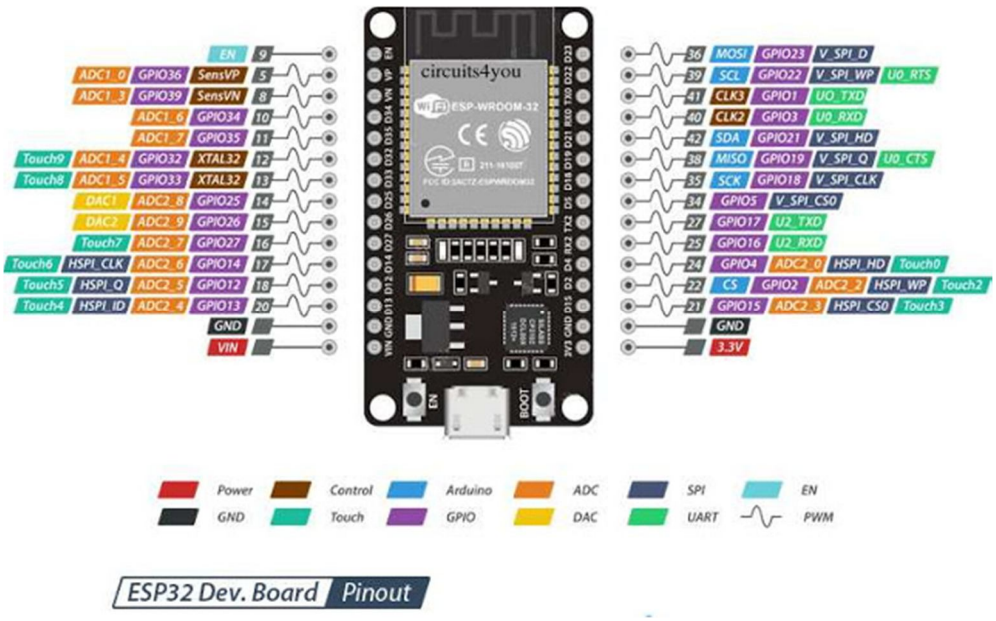
   a) *IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI*

   b) *Secure boot*

   c) *Flash encryption*

   d) *1024-bit OTP, up to 768-bit for customers*

   e) *Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG).*



Gambar 2.2 *Pin Out Module ESP32*
(Sumber: Edukasi Elektronika)

Pada *pin out* tersebut terdiri dari :

a) 18 ADC (*Analog Digital Converter*, berfungsi untuk merubah sinyal analog ke digital)

b) 2 *DAC* (*Digital Analog Converter*, kebalikan dari *ADC*)

c) 16 *PWM* (*Pulse Width Modulation*)

d) 10 Sensor sentuh

e) 2 jalur antarmuka *UART (Universal asynchronous receiver-transmitter)*

f) Pin antarmuka *I2C*, *I2S*, dan *SPI*

F. Bahasa C

Dalam mengontrol *Esp32* kita menggukana bahasa pemrograman yang disebut Bahasa C. Bahasa pemrograman C diciptakan dan dikembangkan oleh *Brian Kernighan dan Denis Ritchie di bell research Labs.* Basa pemrograman C

secara khuusus diciptakan dengan tujuan agar para programmer dapat mengakses seluruh internal register, *I/O slots* dan *abslute addres* dari sebuah komputer. Pada awal tahun 1960-an, sistem operasi komputer mulai menjadi jauh lebih kompleks dari sebelumnya karena adanya pengenalan *multi terminal* dan kemampuan *multi prosesor*. Pada saat itu sistem operasi diciptakan dengan menggunakan bahasa *assembly* (bahasa pemrograman tingkat rendah). Dimana banyak pengembang yang menyadari bahwa satu sistem operasi dapat dikembangkan lebih lagi, tidak cukup hanya menggunakan bahasa *assembly*. Inilah asal mula bahasa C yang di implementasikan pada digital *equipment corporation* PDP-7. Pada pengembangan selanjutnya bahasa pemrograman C digunakan untuk mengimplementasikan sistem operasi linux.

Bahasa C atau C++ adalah suatu bahasa pemrograman. Bahasa C merupakan satu bahasa pemrograman tingkat menengah, maksudnya bahasa C bisa dipelajari dengan lebih mudah karena mudah untuk dipahami dan mempunuyai kemampuan yang tinggi. Bahasa C bisa digunakan untuk merekayasa program untuk segala kebutuhan, baik untuk aplikasi bisnis, matematis atau bahkan game.

Bahasa pemrograman C telah banyak mengalami banyak evolusi sejak awal diciptakannya hingga saat ini. Saat ini, banyak bahasa pemrograman yang merupakan turunan/varian/keluarga dari bahasa C *sepertio C++, Java Script, PHP, Java, Perl* dan lain sebagainya. Bisa juga dibilang bahasa C adalah induk dari bahasa pemrograman saat ini.

Beberapa kelebihan dari bahasa C adalah sebaguai berikut: banyak memiliki operator untuk mengolah/memanipulasi data. Bahasa C termasuk sebagai bahasa yang terstruktur sehingga program dapat lebih mudah difahami uatau dikembangkan. Bahasa C lebih mudah dimengerti karena lebih mirip dengan bahasa manusia. Mengenal data pointer. Bahasa C tersedia hampir disemua jenis komputer. Kode bahasa C sifatnya adalah *portable* dan *fleksibel* untuk semua jenis komputer. Bahasa C hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci. *Proses executable* program bahasa C lebih cepat. Dukungan pustaka yang banyak C adalah bahasayang terstruktur, bahasa C termasuk bahasa tingkat menengah.

Sedangkan kelemahan dari bahasa C adalah: banyaknya operator atau cara penulisan program kadang menimbulkan kebingungan para pemakainya.

Berikut sintaks pada pemrograman bahasa C.

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

1) // (Komentar satu baris)

   Kadag diperlakukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

2) /* /* (Komentar banyak baris)

Jika anda punuya buanyal catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal uyang terletak diantara dua symbol tersebut akan diabaikan oleh program.

3) { } (Kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

4) ; (titik koma)

Setiap baris kode harus diakhiri dengan titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalamkan).

Cara penulisan pada bahasa C

Berikut contoh penulisan komentar pada bahasa C.

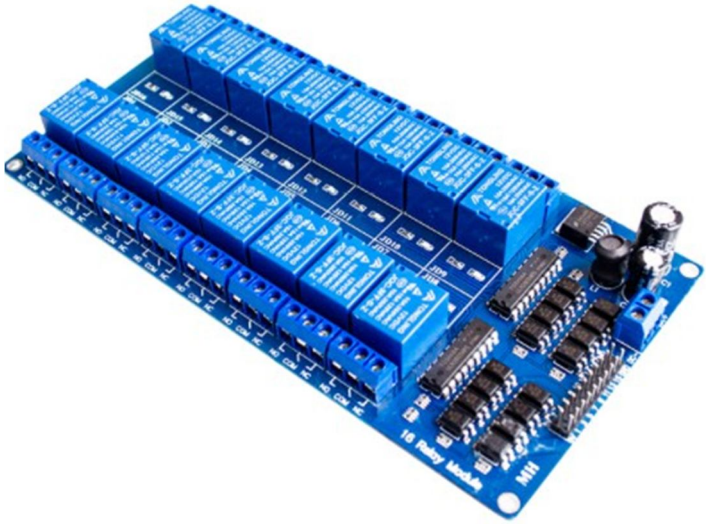
```
1. #include <stdio.h>
2. int main(void)
3. {
4. /* Tampilkan pesan Hello World */
5. printf("Hello, World!\n");
6. return 0;
7. }
```

G. *Relay*

Relay adalah salah satu komponen elektronika yang berfungsi untuk menyambung dan memutuskan arus listrik dalam sebuah rangkaian. Karena fungsi relay tersebut, itulah mengapa komponen yang satu ini juga disebut sebagai

saklar. Hal ini karena komponen yang satu ini berguna untuk membuka dan menutup aliran arus listrik.

Pada *relay* ini terdiri dari dua komponen utama. Yaitu komponen kumparan atau elektromagnetik dan juga kontak point atau saklar.



Gambar 2.3 *Relay 16 Channel*
(Sumber: Shopee)

Kegunaan relay dalam sebuah rangkaian elektronik cukup vital. Berikut ini 6 fungsi yang perlu anda ketahui:

1) Mengendalikan sirkuit dengan tegangan tinggi dengan menggunakan derma signal bertegangan rendah.
2) Memperkecil terjadinya penurunan tegangan.
3) Menjalankan fungsi logika atau *logic function.*
4) Memberikan fungsi penundaan waktu atau *delay time function.*
5) Melindungi komponen elektronika dari bahaya konsleting.
6) Memperingkas rangkaian elektronika. Jadi arus listrik dapat langsung menuju beban tanpa harus melewati komponen elektronika yang lainnya.

Jenis-jenis relay dapat dibedakan atau dikelompokkan menjadi dua yaitu berdasarkan posisi awal atau kontak pointnya atau juga berdasarkan jumlah *Phole* dan *Throw* yang dimilikinya.

Berikut perbedaan dari keduanya :

1) Jenis *Relay* Berdasarkan Posisi Awalnya

   Jenis Relay yang pertama yaitu *Relay* berdasarkan posisi awal dari kontak pointnya. Berdasarkan posisi awal dari kontak pointnya, *Relay* dibagi menjadi dua yaitu :

   a) *Relay Type Normally Close* (NC)

      Tipe *Relay* yang satu ini merupakan *Relay* yang pada kondisi awal sebelum diaktifkan berada pada posisi terhubung atau menutup (*close*).

   b) *Relay Type Normally Open* (NO)

      Tipe *Relay Normally Open* adalah merupakan *relay* yang pada kondisi awal sebelum diaktifkan berada pada posisi terputus atau terbuka (*open*).

2) Jenis *Relay* Berdasarkan Jumlah *Pole* dan *Throw*

   Jenis *relay* yang kedua adalah dibedakan berdasarkan jumlah *pole* (banyaknya kontak yang dimiliki) dan *throw* (banyaknya kondisi berdasarkan kontak pointnya). Berdasarkan *Pole* dan *Throw*, Relay dikelompokan menjadi beberapa yaitu :

   a) *Relay Tipe Single Pole Single Throw* (SPST)

      Jenis *Relay tipe Single Pole Single Throw* adalah salah satu jenis *Relay* yang memiliki empat kaki terminal. Yang mana dua kaki terminal berfungsi

sebagai kontak *point* (saklar) dan dua kaki terminal yang lainnya memiliki fungsi sebagai kumparan elektromagnet.

b) *Relay Type Single Pole Double Throw* (SPDT)

Relay tipe yang satu ini didesain dengan memiliki lima kaki terminal. Dimana tiga kaki terminal ini digunakan sebagai kontak point (saklar) sedangkan dua kaki terminal yang lainnya memiliki fungsi sebagai kumparan elektromagnet.

c) *Relay Type Double Pole Single Throw* (DPST)

Tipe *relay* yang satu ini memiliki karakteristik yaitu memiliki enam kaki terminal. Dimana empat kaki terminal berfungsi sebagai saklar dan dua kaki terminal berfungsi sebagai kumparan elektromagnet. Pada *relay* jenis ini, empat kaki terminalnya terdiri dari dua pasang saklar *single pole double throw.*

d) Relay Type Double Pole Double Throw

Relay tipe *Double Pole Double Throw* ini di desain dengan delapan buah terminal. Dimana pada *Relay* tipe ini 6 terminalnya memiliki fungsi sebagai kontak *point* (saklar). Sedangkan dua terminal lainnya berfungsi sebagai kumparan elektromagnet. Pada *Relay* jenis ini, enam pasang terminal yang berfungsi sebagai kontak *point* ini terdiri dari dua pasang saklar *single pole double throw.*

H. Lampu Lombok

Lampu lombok merupakan komponen bahlom *led* yang dilengkapi dengan fiting lampu sebagai tempat arus listrik. Lampu lombok ini beragam warna sehingga nampak indah dan lampu lombok ini semikonduktor yang memancarkan

cahaya ketika di aliri. Warna-warna cahaya yang dipancarkan oleh lampu tergantung pada jenis bahan semikonduktor yang dipergunakannya.

Bentuk lampu lombok itu kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika.



Gambar 2.4 Lambu Lombok
(Sumber: Shopee)

Spesifikasi Lampu :

1) Daya : 5 Watt
2) Fitting : E12
3) Warna : Biru, Kuning Hijau, Merah
4) Dimensi : 5,5 cm x 2,5 cm

I. Catu Daya (*Power Supply*)

Rangkaian *Power Supply merupakan* alat yang berfungsi menyuplai daya listrik ke peralatan elektronik. Daya yang masuk umumnya berasal dari konversi tegangan PLN 220V menjadi tengan *DC* antara 5V sampai 12V. Namun beberapa tipe *Power supply* ada yang mencapai 70V DC.



Gambar 2.5 *Power Supply 5V*
(Sumber: Alibaba)

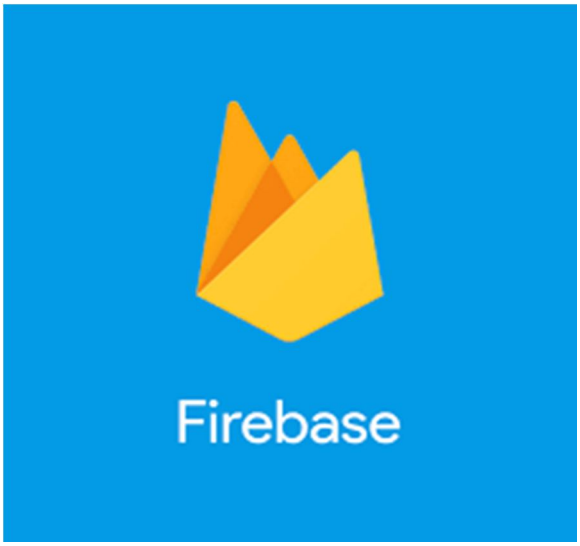
Spesifikasi *Power Supply* :

1) Model : S-25-5
2) AC Input : 110/220V±15%
3) DC Output : 5V 5A

Para teknisi lebih sering menyebut *Power Supply* sebagai Catu daya. Namun untuk sekarang lebih dikenal *power supply* sebagai Adaptor. Tapi pada dasarnya mempunyai prinsip yang sama saya, yaitu mengatur arus yang masuk ke perangkat elektronik dengan mengonversi tegangan 220V dari PLN.

J. *Firebase*

Firebase adalah suatu layanan *Google* yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya *Firebase*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua *fitur* yang menarik dari *Firebase* yaitu *Firebase Remote Config dan Firebase RealTime Database.*



Gambar 2.6 *Firebase*
(Sumber: www.firebase.google.com)

Realtime Database adalah *database NoSQL*, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan *database* terkait. *API Realtime Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman *realtime* yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut. Berikut ini bagaimana proses penginstalan dan persiapan *firebase* ke aplikasi *Android*:

1. Menghubungkan aplikasi ke *Firebase*

   Tambahkan *Firebase* ke *project Android* yang mana dalam file *build.gradle level project*, pastikan untuk memasukkan *repository Maven Google* dibagian *buildscript* dan *allprojects*.

2. Menambahkan Realtime *Database* ke aplikasi

Menambahkan dependensi untuk *Realtime Database* ke file *build.gradle* level aplikasi dengan *script* aplikasi:

```
implementation 'com.google.firebase:firebase-database:19.1.0'
```

3. Mengkonfigurasi Aturan *Realtime Database*

Realtime Database menyediakan bahasa aturan deklaratif sehingga programmer dapat menentukan cara data disusun, *indeks*, serta kapan data tersebut dapat dibaca dan ditulisi. Secara *default*, akses baca dan tulis ke *database* dibatasi, sehingga hanya pengguna yang terautentikasi yang dapat membaca atau menulis data. Untuk memulai tanpa menyiapkan *Autentikasi*, serta dapat mengonfigurasi aturan agar dapat diakses oleh publik. Tindakan ini membuat *database* terbuka bagi siapa saja, termasuk orang yang tidak menggunakan aplikasi. Oleh karena itu, pastikan kembali untuk membatasi *database* setelah menyiapkan *autentikasi*.

4. Menulis ke *database* Anda

Ambil *instance database* menggunakan *getInstance ()* dan rujukan lokasi tempat dimana ingin menulis.

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");

myRef.setValue("Hello, World!");
```

Sehingga bisa menyimpan berbagai jenis data ke *database* dengan cara ini, termasuk objek *Java*. Ketika menyimpan sebuah objek, *respons* dari setiap *getter* akan disimpan sebagai turunan dari lokasi ini.

5. Membaca dari *database* Anda

Agar data aplikasi diupdate secara *realtime*, maka perlu untuk menambahkan *Value Event Listener* ke referensi yang baru saja dibuat. Metode *on Data Change* () di class ini dipicu satu kali saat pemroses ditambahkan dan dipicu lagi setiap kali terjadi perubahan data, termasuk turunannya.

```
// Read from the database
myRef.addValueEventListener(newValueEventListener(){
    @Override
    publicvoid onDataChange(DataSnapshot dataSnapshot){
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        String value = dataSnapshot.getValue(String.class);
        Log.d(TAG,"Value is: "+ value);
    }
```

K. *Mit App Inventor*

Sistem berbasis *web* dimana aplikasi *Android* dapat digunakan tanpa perlu tahu bagaimana cara meng-*code*-nya. Sistem ini telah dihentikan oleh *google* tapi dirilis kembali oleh google sebagai proyek *open-source* dan saat ini dikelola oleh *Massachusetts Institute of Technology (MIT).*

Dengan *app inventor*, pengguna bisa melakukan pemrograman komputer untuk menciptakan aplikasi perangkat lunak dengan sistem operasi berbasi

android. *App inventor* ini berbasis *visual block* programming karena memungkinkan pengguna bisa menggunakan, melihat, menyusun dan men-*drag and drops block* yang merupakan simbol perintah dan fungsi *event handler* untuk menciptakan sebuah aplikasi yang bisa berjalan di sistem *android*.

MIT App Inventor merupakan *platform* untuk memudahkan proses pembuatan aplikasi sederhana tanpa harus mempelajari atau menggunakan bahasa pemrograman yang terlalu banyak. Kita dapat mendesain aplikasi *android* sesuai keinginan dengan menggunakan berbagai macam layout dan komponen yang tersedia.

App Inventor memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi *Android*. *App Inventor* menggunakan antarmuka grafis, serupa dengan antarmuka pengguna pada *Scratch* dan *StarLogo TNG,* yang memungkinkan pengguna untuk men-*drag-and-drop objek* visual untuk menciptakan aplikasi yang bisa dijalankan pada perangkat *Android*. Dalam menciptakan *App Inventor*, *Google* telah melakukan riset yang berhubungan dengan komputasi edukasional dan menyelesaikan lingkungan pengembangan *online Google*

L. Pengenalan *Android*

Irawan **(2012:2)** menyatakan, bahwa *android* merupakan sebuah sistem operasi yang berbasis *linux* untuk perangkat *portable* seperti *smartphone* dan komputer *tablet*. *Android* menyediakan *platform* terbuka (*open source*) bagi

programmer untuk mengembangkan aplikasi sendiri pada berbagai prangkat dengan sistem *android*.

Supardi **(2012:3)** menyatakan, bahwa sejarah awal *android* berawal dari sebuah perusahaan software kecil yang didirikan pada bulan oktober 2003 di *Palo Alto, California USE*. Didirikan oleh bebertapa senior di beberapa prusahaan yang berbasis *IT* dan *Communication,* yakni Andy Rubin, Rich Miner, Nick Sear, dan Cris White. Pada bulan Agustus 2005, *android* diakuisisi oleh *Google Inc*, seluruh sahamnya dibeli oleh *Google*. Untuk pengembangan lanjutan *android*, dibentuklah *OHA (open Handset Alliance)* konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi. Termasuk diantaranya adalah *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*. Pada saat rilis perdana Android, 5 November 2007, android bersama *OHA* menyatakan mendukung pengembangan standar terbuka pada perangkat *smartphone*. *Google* mengajukan hak paten aplikasi ponsel *android* pertama pada september 2007. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android* ARM *Holding, Atheros Communication,* diproduksi oleh *Asustek Computer Inc, Garmin Ltd, Softbank, Sonny Ericsson, Toshiba Corp, dan Vodafone Group Plc.*

M. *Flow Map* (Bagan Alir)

Menurut Abdul Kadir **(2009:10)** bahwa *Flowmap* atau bagan alir adalah bagan yang menunjukan aliran di dalam program atau prosedur sistem secara logika. *Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses

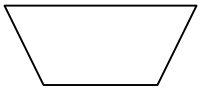
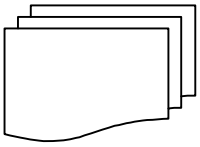
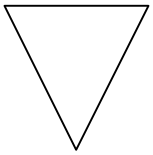
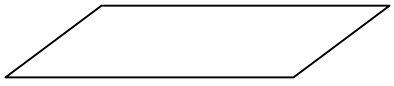
maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowmap* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi. Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:

1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukan dengan jelas.
3. Harus ditunjukan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam *flowmap* harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Adapun simbol-simbol yang sering digunakan dalam *flowmap* dapat dilihat pada tabel 2.3 berikut ini:

Tabel 2.1 : Simbol dan Keterangan *Flowmap*

Simbol	**Deskripsi**
	Simbol yang digunakan untuk menunjukkan awal atau akhir dari suatu proses

Simbol	Deskripsi
	Menunjukkan dokumen input dan output baik untuk proses manual mekanik atau komputer
	Menunjukkan pekerjaan manual
	Menunjukkan multi dokumen
	Pengarsipan Data
	Menunjukkan Proses
	Simbol i*nput/output* digunakan untuk mewakili data *input/output*

1. *UML (Unified Modeling Language)*

Defenisi *UML*, Menurut Windu Gata **(2013)** Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). *UML*

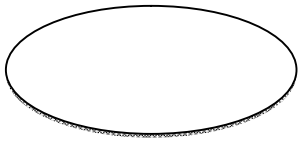
adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan *system.*

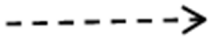
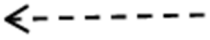
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut :

a) *Use Case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol- simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel 2.2 *Simbol Use Case*

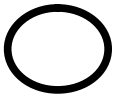
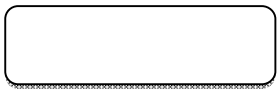
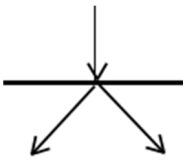
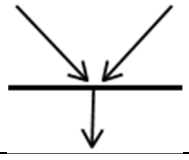
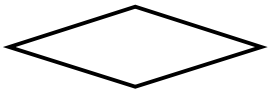
Gambar	**Keterangan**
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.

Gambar	Keterangan
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Mengidentifikasikan aktor harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki *control* terhadap *use*
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah Pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

b) *Activity* Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam activity diagram yaitu:

Tabel 2.3 *Simbol Activity Diagram*

Gambar	Keterangan
	*Start point,* diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point,* akhir aktifitas.
	*Activites,* menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane,* pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

c) *Class* Diagram

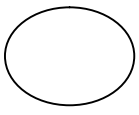
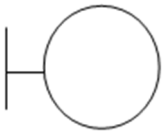
Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

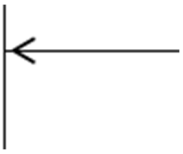
Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class* diagram secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes)*, Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

d) *Sequense* Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence* diagram yaitu :

Tabel 2.4 *Simbol Sequense* Diagram

Gambar	Keterangan
	*EntityClass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *formentry* dan *form* cetak.

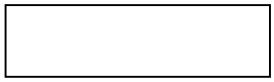
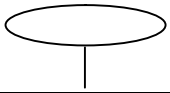
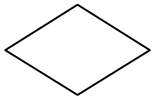
Simbol	Deskripsi
	*Control class,* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message,* simbol mengirim pesan antar *class.*
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation.*

2. *Entity Relationalship* Diagram

Abdul Kadir, **(2009:14)** bahwa *Entity Relationship Diagram* (E-R Diagram) adalah suatu model relasi yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Diagram E-R merupakan model E-R yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi atribut-atribut yang mempersentasikan seluruh fakta dari dunia nyata yang kita tinjau, dan dapat digambarkan dengan lebih sistematis.

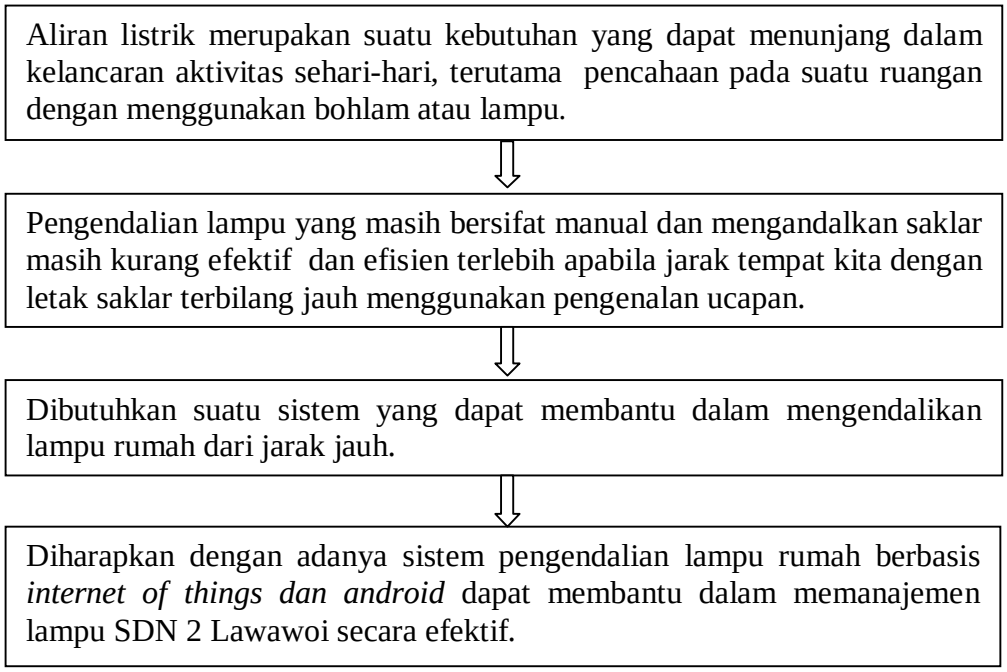
Notasi-notasi dalam E-R Diagram yang dapat digambarkan pada tabel 2.5. berikut ini:

Tabel 2.5 Simbol dan keterangan E-R Diagram

Simbol	Deskripsi
	Menunjukkan entitas yang terhubungan dengan sistem
	Menunjukkan atribut yang dimiliki oleh entitas
	Menunjukkan relasi antar entitas
	Menunjukkan *link*

N. Kerangka Pikir

Untuk lebih memperjelas kerangka berpikir yang akan disajikan, maka akan digambarkan sebagai berikut:



Gambar 2.7 Diagram Kerangka Berpikir

BAB III

METODOLOGI PENELITIAN

A. Waktu dan Lokasi Penelitian

Lokasi penelitian dilaksanakan di SDN 2 Lawawoi yang beralamat di Jln. Jenderal Sudirman No.08 Kelurahan Bangkai, Kecamatan Watang Pulu, Kabupaten Sidenreng Rappang. Waktu penelitian dilakukan mulai bulan September 2019 sampai dengan bulan November 2019.

B. Jenis Penelitian

Dalam menyelesaikan skripsi ini, jenis penelitian yang dilakukan adalah sebagai berikut :

1. Penelitian pustaka (*Library Research*), yaitu penelitian yang di lakukan dengan menggunakan beberapa buku sebagai referensi untuk penulisan.
2. Penelitian lapangan (*Field Research*), yaitu penelitian dilakukan dengan cara mengamati proses kerja dari penggunaan saklar dalam pemanfaatan lampu.

C. Teknik Pengumpulan Data

Teknik pengumpulan data dilakukan dengan cara :

1. Observasi

   Penelitian dilakukan dengan cara mengumpulkan data-data yang akan dijadikan bahan dasar dalam perancangan sistem pengendalian lampu *internet of things* dan *android*

2. Wawancara

   Penelitian dilakukan dengan proses tanya jawab kepada beberapa pihak untuk mengetahui permasalahan yang akan dibahas

D. Alat dan Bahan Penelitian

1. Alat

   Alat yang digunakan dalam penelitian berupa :

   a) Perangkat Lunak (*Software*)

      Perangkat lunak (*software*) yang digunakan untuk membangun sistem kontrol lampu berbasis *Internet of Things* dan *android* pada SDN 2 Lawawoi ini dapat dilihat pada spesifikasi sebagi berikut :

      1) *Windows 10 Ultimate* sebagai sistem operasi
      2) *Arduino IDE*
      3) *Web Browser Google Chrome*

   b) Perangkat Keras (*Hardware*)

      Perangkat keras (*hardware*) yang digunakan untuk membangun sistem kontrol lampu berbasis *Internet of Things* dan *android* pada SDN 2 Lawawoi ini dapat dilihat pada spesifikasi sebagi berikut :

      1) Laptop Acer Aspire E5-475G dengan spesifikasi sebagai berikut:
         a) *Processor Intel(R) Core(TM)* i5-7200U CPU @2.50GHz
         b) *RAM* 8 GB
         c) *Harddisk* 500 (Lima Ratus) *GB*
      2) *Smartphone Android*

3) *Esp32*

4) *Relay*

5) *Lampu Lombok*

6) *Power Supply*

E. Tahapan Penelitian

Beberapa tahapan yang dilaksanakan selama penelitian, sebagai berikut :

a) Pengumpulan Data

Pada tahapan ini d ilakukan pengumpulan data berupa arsip-arsip dan bahan-bahan pustaka yang berhubungan dengan sistem yang akan dirancang.

b) Analisis Sistem

Melakukan analisis sistem yang sedang berjalan saat itu dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan, hambatan yang terjadi, dan kebutuhan-kebutuhan yang diharapkan.

c) Desain Sistem

Setelah melakukan pengumpulan data, pada tahap ini dilakukan perancangan konsep sebuah sistem yang dibutuhkan.

d) Coding

Tahap penulisan kode (*listing*) program sistem.

e) Pengujian Sistem

Tahapan dimana siste yang telah dirancang akan diuji keakuratannya, apakah sudah bebas dari kesalahan-kesalahan sesuai dengan dengan metode pengujian *black box*.

f) Implementasi

Implementasi sistem adalah tahap meletakkan aplikasi agar dapat beroperasi.

F. Jenis data

Jenis data yang dikumpulkan dalam penelitian ini adalah data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

1. Data Primer

   Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber-sumber yang berada di lapangan yang berasal dari wawancara dengan pihak-pihak terkait dengan penelitian ini.

2. Data Sekunder

   Data Sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Penelitian medapatkan data yang sudah jadi yang dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial.

G. Metode Pengujian

Metode pengujian yang dilakukan pada penelitian ini adalah pengujian yang dipakai untuk mengetahui kinerja prototype pada saat melakukan proses kontrol. Metode pengujian yang dilakukan terdiri dari :

1. *Black Box Testing*

   Sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak

tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, *tester* menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

Kelebihan *Black Box Testing* yaitu:

a) Efisien untuk segmen kode besar
b) Akses kode tidak diperlukan
c) Pemisahan antara perspektif pengguna dan pengembang

Kelemahan *Black Box Testing* yaitu:

a) Cakupan terbatas karena hanya sebagian kecil dari skenario pengujian yang dilakukan
b) Pengujian tidak efisien karena keberuntungan *tester* dari pengetahuan tentang perangkat lunak internal

2. *White Box Testing*

White Box Testing merupakan metode pengujian perangkat lunak di mana struktur internal diketahui untuk menguji siapa yang akan menguji perangkat lunak. Pengujian ini membutuhkan pengetahuan internal tentang kemampuan sistem dan pemrograman.

Kelebihan *White Box Testing* yaitu:

a) Efisien dalam menemukan kesalahan dan masalah
b) Diperlukan pengetahuan tentang internal perangkat lunak yang sedang diuji bermanfaat untuk pengujian menyeluruh
c) Memungkinkan menemukan kesalahan tersembunyi
d) Membantu mengoptimalkan kode

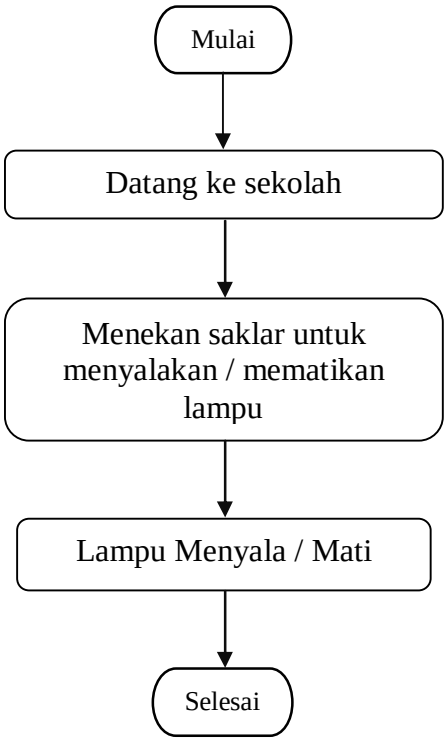
Kelemahan *White Box Testing* yaitu:

a) Membutuhkan pengetahuan tingkat tinggi dari perangkat lunak internal yang sedang diuji
b) Membutuhkan akses kode

BAB IV

HASIL DAN PEMBAHASAN

H. Rancangan sistem yang berjalan

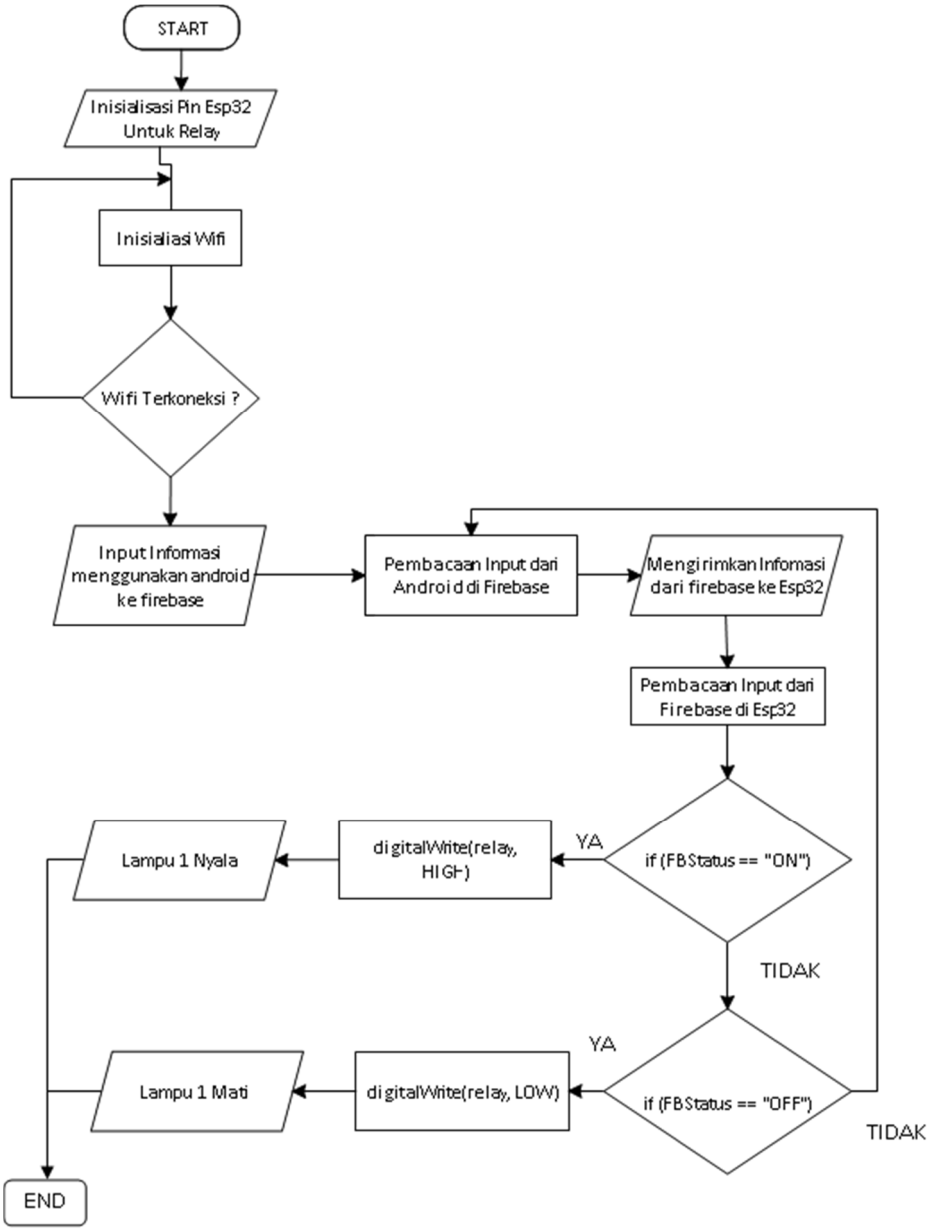


Gambar 4.1 Sistem yang berjalan

Pada gambar diatas menjelaskan bagaimana proses secara umum ketika menyalakan atau mematikan lampu secara manual yang mana harus dilakukan dalam wilayah sekolah.

I. Rancangan Sistem yang diusulkan

Sistem dimulai dari mengoprasikan komponen alat yang saling terhubung pada *prototype* sistem kontrol lampu berbasis *internet of things* dan *android* pada SDN 2 Lawawoi yaitu mengubungkan *Arduino uno, Power Supply, relay 16 channel, Esp32.*

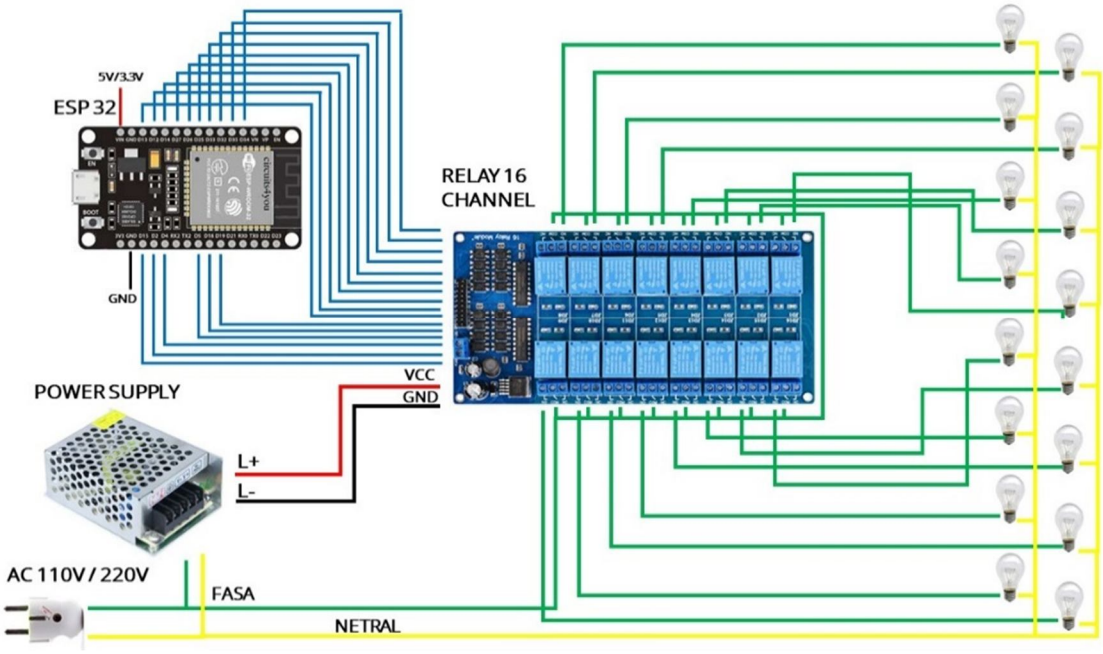


Gambar 4.2 *Flowchart* sistem Kontrol Lampu yang diusulkan

Penjelasan dari *flowchart* tersebut adalah pertama melakukan inisialisasi pin *Esp32* yang terhubung pada *relay* yang kemudian dilanjutkan inisialisasi jaringan, apabila jaringan telah terkoneksi maka *Esp32* akan menerima data dari *firebase*. Pada saat mengirim perintah melalui aplikasi khusus di *Android*, maka perintah tersebut akan dikelola di firebase, dan kemudian akan diteruskan ke mikrokontroler *Esp32* untuk eksekusi. Jika perintah yang diterima *ON* maka lampu akan menyala. Dan jika perintah yang diterima *OFF* maka lampu akan mati.

J. Rancangan Perangkat Keras (*Hardware*)

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektroniknya. Rancangan ini akan menjelaskan alat yang digunakan dan memilih komponen yang tepat.

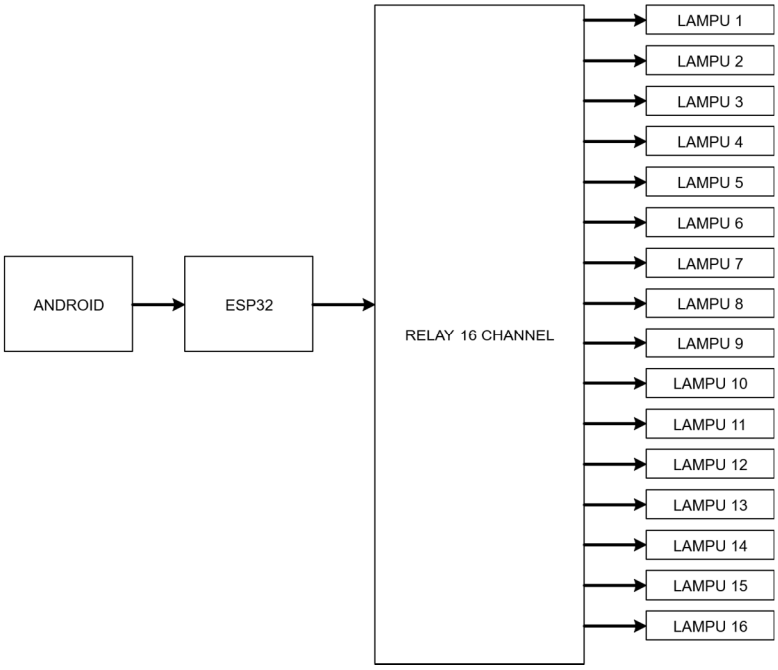


Gambar 4.3 Rangkaian Alur Pengkabelan

Pada gambar diatas merupakan alur pengkabelan atau rangkaian perancangan sistem kontrol lampu berbasis *Internet of Things* dan *Android* pada SDN 2 Lawawoi. Pada dasarnya rancangan *prototype* sistem kontrol lampu berbasis *internet of things* dan *android* pada SDN 2 Lawawoi ini mempunyai bagian *hardware* yang terdiri dari : *Esp32, relay 16 channel,* Lampu lombok, dan *Power supply*.

K. Implementasi Perangkat Keras (*Hardware*)

Desain diagram blok terdiri dari blok input disini menggunakan smartphone android yang terkoneksi internet yang sudah terinstal aplikasi kontrol lampu SDN 2 Lawawoi, smartphone android ini digunakan untuk mengirimkan perintah untuk menyalakan dan mematikan Lampu yang kemudian dikirim melalui *Google Firebase* ke Esp32 yang juga sudah terkoneksi dengan internet.

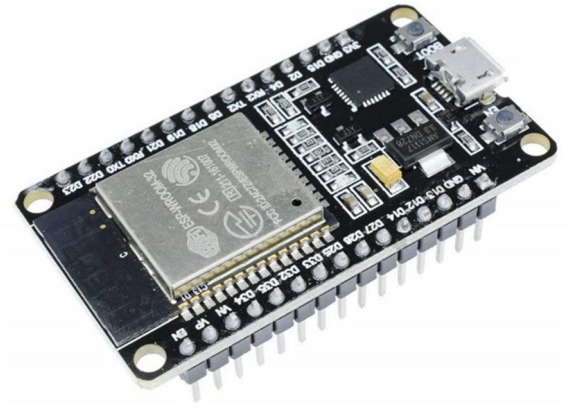


Gambar 4.4 Blok diagram sistem

Adapun uraian masing-masing *hardware* yang digunakan sebagai berikut :

1. *Esp32*

   ESP32 adalah *mikrokontroler* yang dikenalkan oleh *Espressif System* merupakan penerus dari *mikrokontroler ESP8266*. Pada *mikrokontroler* ini sudah tersedia modul *WiFi* dalam *chip* sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. terlihat pada gambar di atas merupakan *pin out* dari *ESP32*. *Pin* tersebut dapat dijadikan *input* atau *output* untuk menyalakan *LCD*, lampu, bahkan untuk menggerakan motor *DC*.



Gambar 4.5 *Esp32*
(Sumber: Tokopedia)

Tabel 4.1 Spesifikasi *Esp32*

No.	Nama	Keterangan
1.	*Mikrokontroler*	*Esp32*
2.	Tegangan Pengoperasian	3.3V
3.	Batas Tegangan *Input*	0.75 - 3.3V
4.	Pin I/O Digital	36 (18/2)
5.	Pin Digital *PWM*	16 (kecuali GPIO 34 hingga 39)
6.	Pin *ADC*	18

No.	Nama	Keterangan
7	Pin *DAC*	2 (GPIO 25 & 26)
8.	Arus DC Tiap Pin I/O	12 mA
9.	*Arus DC untuk pin 3.3V*	*50 mA*
10.	*Flash Memory*	*16 MB (Esp32)*
11.	*SRAM*	*520 KB (Esp32)*
12.	*EEPROM*	*512 byte (Esp32)*
13.	*Clock Speed*	*40 MHz*
14.	Panjang	*68.6 mm*
15.	Lebar	*53.4 mm*
16.	Berat	*25 g*

Satu perbedaan antara *ESP32* dengan *ESP8266* adalah pada bagian prosesornya. *ESP32* sudah *Dual-Core 32 bit*, jelas lebih cepat *ESP32* secara kinerja. Selain itu modul ini juga mempunyai *bluetooth* , satu fitur yang tidak ada di *ESP8266*.

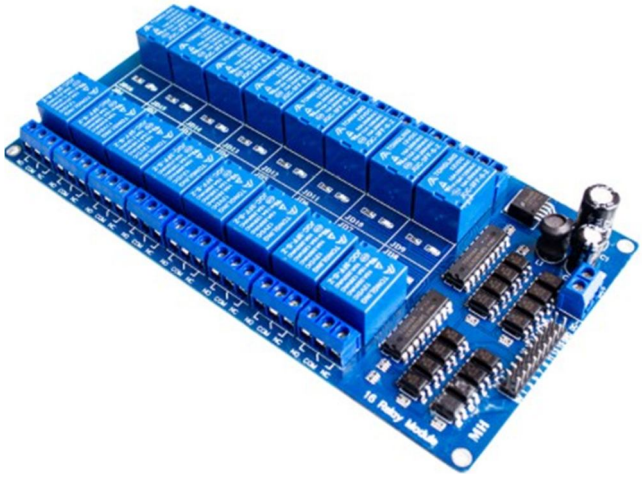
Tabel 4.2 Pengunaan pin *Esp32*

No.	Pin/Soket *Esp32*	Keterangan
1.	GND	Dihubungkan ke V- *Power Supply*
2.	VCC	Dihubungkan ke V+ *Power Supply*
3.	GPIO 15	Dihubungkan ke IN3 *Relay*
4.	GPIO 2	Dihubungkan ke IN1 *Relay*
5.	GPIO 4	Dihubungkan ke IN5 *Relay*
6.	GPIO 16	Dihubungkan ke IN16 *Relay*
7.	GPIO 17	Dihubungkan ke IN13 *Relay*
8.	GPIO 5	Dihubungkan ke IN11 *Relay*
9.	GPIO 18	Dihubungkan ke IN9 *Relay*

10.	GPIO 19	Dihubungkan ke IN10 *Relay*
11.	GPIO 21	Dihubungkan ke IN4 *Relay*
12.	*GPIO* 13	Dihubungkan ke IN2 *Relay*
13.	*GPIO* 14	Dihubungkan ke IN6 *Relay*
14.	*GPIO* 27	Dihubungkan ke IN14 *Relay*
15.	*GPIO* 26	Dihubungkan ke IN15 *Relay*
16.	*GPIO* 25	Dihubungkan ke IN12 *Relay*
17.	*GPIO* 33	Dihubungkan ke IN7 *Relay*
18.	*GPIO* 32	Dihubungkan ke IN8 *Relay*

2. *Relay* 16 *Channel*

   Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni *Elektromagnet* (*Coil*) dan Mekanikal (seperangkat Kontak Saklar/*Switch*). *Relay* menggunakan Prinsip *Elektromagnetik* untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *Relay* yang menggunakan *Elektromagnet 5V* dan 50 *mA* mampu menggerakan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220*V* 2*A*.



Gambar 4.6 Relay 16 Channel

(Sumber: Shopee)

Tabel 4.3 Pengunaan pin *Relay*

No.	Pin Relay 16 Channel	Keterangan
1.	*GND*	Dihubungkan ke V- *Power Supply*
2.	*VCC*	Dihubungkan ke V+ *Power Supply*
3.	*Channel IN1*	Dihubungkan ke *GPIO 2 Esp32*
4.	*Channel IN2*	Dihubungkan ke *GPIO 13 Esp32*
5.	*Channel IN3*	Dihubungkan ke *GPIO 15 Esp32*
6.	*Channel IN4*	Dihubungkan ke *GPIO 21 Esp32*
7.	*Channel IN5*	Dihubungkan ke *GPIO 4 Esp32*
8.	*Channel IN6*	Dihubungkan ke *GPIO 14 Esp32*
9.	*Channel IN7*	Dihubungkan ke *GPIO 33 Esp32*
10.	*Channel IN8*	Dihubungkan ke *GPIO 13 Esp32*
11.	*Channel IN9*	Dihubungkan ke *GPIO 18 Esp32*
12.	*Channel IN10*	Dihubungkan ke *GPIO 19 Esp32*
13.	*Channel IN11*	Dihubungkan ke *GPIO 5 Esp32*
14.	*Channel IN12*	Dihubungkan ke *GPIO 25 Esp32*
15.	*Channel IN13*	Dihubungkan ke *GPIO 17 Esp32*

16.	*Channel IN14*	Dihubungkan ke *GPIO 27 Esp32*
No.	**Pin/Soket *Esp32***	**Keterangan**
17.	*Channel IN15*	Dihubungkan ke *GPIO 25 Esp32*
18.	*Channel IN16*	Dihubungkan ke *GPIO 16 Esp32*

Seperti yang telah di jelaskan tadi bahwa *relay* memiliki fungsi sebagai saklar elektrik, namun jika di aplikasikan ke dalam rangkaian elektronika, relay memiliki beberapa fungsi yang cukup unik. Berikut beberapa fungsi saat di aplikasikan ke dalam sebuah rangkaian elektronika.

a) Mengendalikan sirkuit tegangan tinggi dengan menggunakan bantuan signal tegangan rendah.
b) Menjalankan *logic function* atau fungsi logika.
c) Memberikan *time delay function* atau fungsi penundaan waktu.
d) Melindungi motor atau komponen lainnya dari korsleting atau kelebihan tegangan.

3. Catu Daya (*Power Supply*)

Rangkaian *Power Supply merupakan* alat yang berfungsi menyuplai daya listrik ke peralatan elektronik. Daya yang masuk umumnya berasal dari konversi tegangan PLN 220V menjadi tengan *DC* antara 5V sampai 12V. Namun beberapa tipe *Power supply* ada yang mencapai 70V DC.



Gambar 4.7 Power Supply 5V

(Sumber: Shopee)

Tabel 4.4 Pengunaan *Power Supply*

No.	Pin/Soket *Esp32*	Keterangan
1.	*V+*	Dihubungkan ke VCC *Esp32 & Relay*
2.	*V-*	Dihubungkan ke GND *Esp32 & Relay*
3.	*L (AC IN)*	Dihubungkan ke Lampu, *Relay* dan Arus Listrik
4.	*N (AC IN)*	Dihubungkan ke Lampu dan Arus Listrik

4. Lampu Lombok

Lampu lombok adalah lampu berukura kecil yang dilengkapi dengan fiting lampu sebagai tempat arus listrik. Lampu ini memiliki berbagai macam warna sehingga terlihat indah. Lampu kecil ini semikonduktor yang memancarkan cahaya ketika di aliri.

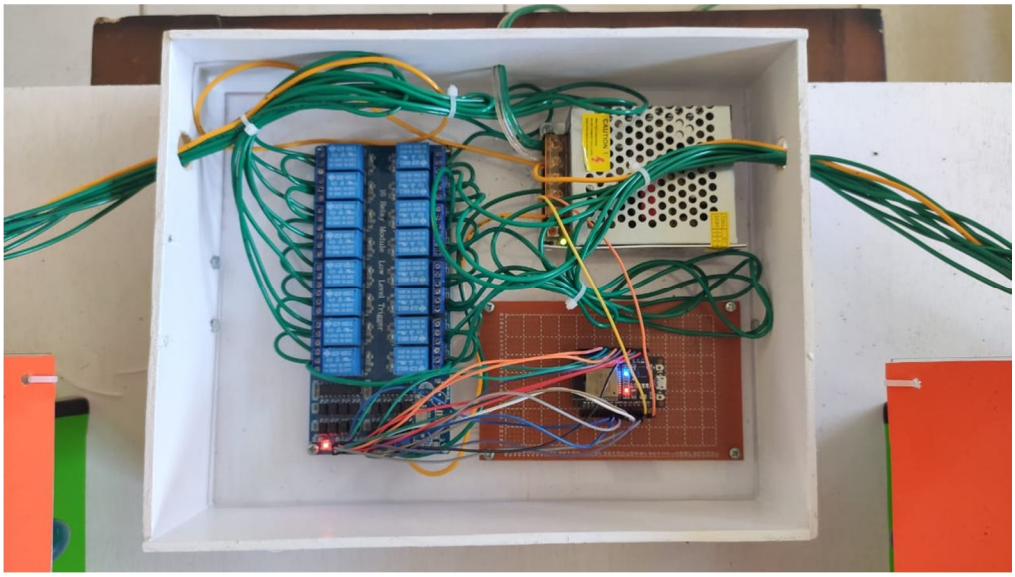


Gambar 4.8 Lambu Lombok
(Sumber: Shopee)

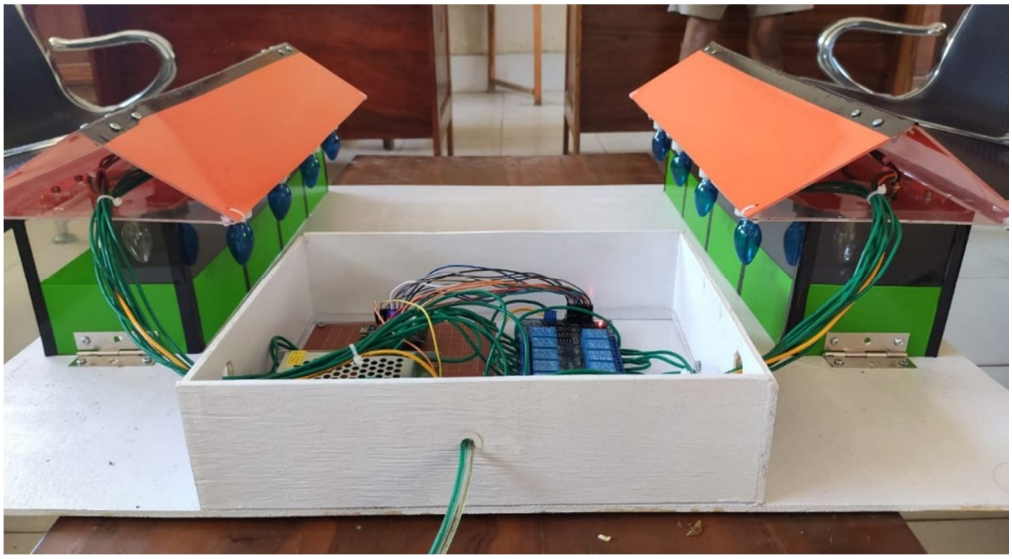
Adapun arus kabel yang digunakan lampu lombok yaitu:

Tabel 4.5 Penggunaan arus kabel lampu lombok

No.	Soket lampu lombok	Keterangan
1.	Lampu Lombok	Dihubungkan ke arus listrik PLN
2.	Lampu Ruang Kantor	Dihubungkan ke IN3 *Relay*
3.	Lampu Ruang Toilet	Dihubungkan ke IN1 *Relay*
4.	Lampu Ruang Kelas 6	Dihubungkan ke IN5 *Relay*
5.	Lampu Ruang Kelas 5	Dihubungkan ke IN16 *Relay*
6.	Lampu Ruang Kelas 4	Dihubungkan ke IN13 *Relay*
7.	Lampu Ruang Kelas 3	Dihubungkan ke IN11 *Relay*
8.	Lampu Ruang Kelas 2	Dihubungkan ke IN9 *Relay*
9.	Lampu Ruang Kelas 1	Dihubungkan ke IN10 *Relay*
10.	Lampu Teras Kantor	Dihubungkan ke IN4 *Relay*
11.	Lampu Teras Toilet	Dihubungkan ke IN2 *Relay*
12.	Lampu Teras Kelas 6	Dihubungkan ke IN6 *Relay*
13.	Lampu Teras Kelas 5	Dihubungkan ke IN14 *Relay*
14.	Lampu Teras Kelas 4	Dihubungkan ke IN15 *Relay*
15.	Lampu Teras Kelas 3	Dihubungkan ke IN12 *Relay*
16.	Lampu Teras Kelas 2	Dihubungkan ke IN7 *Relay*
17.	Lampu Teras Kelas 1	Dihubungkan ke IN8 *Relay*



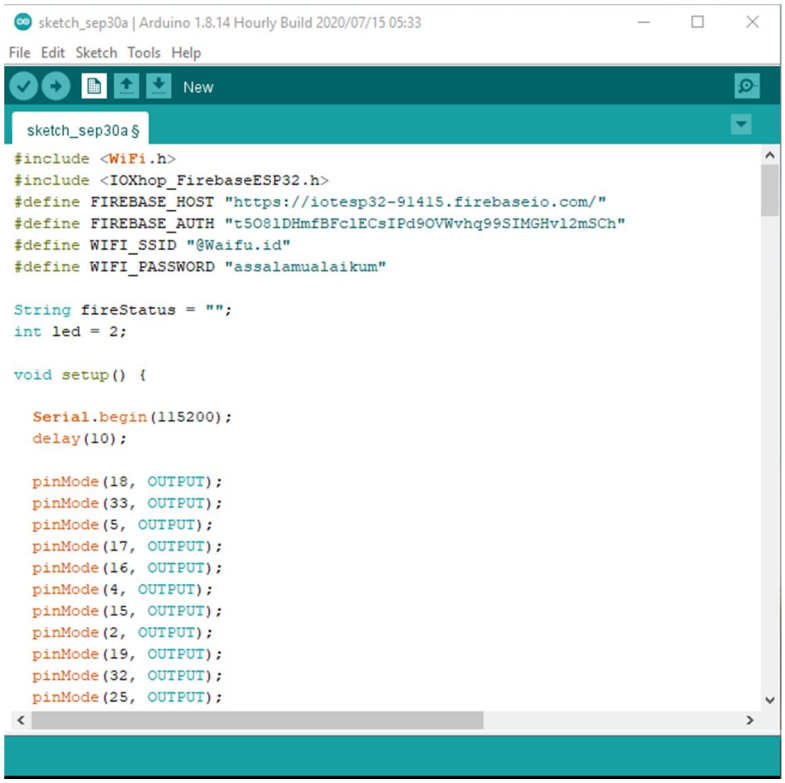
Gambar 4.9 Bentuk dalam *Box*

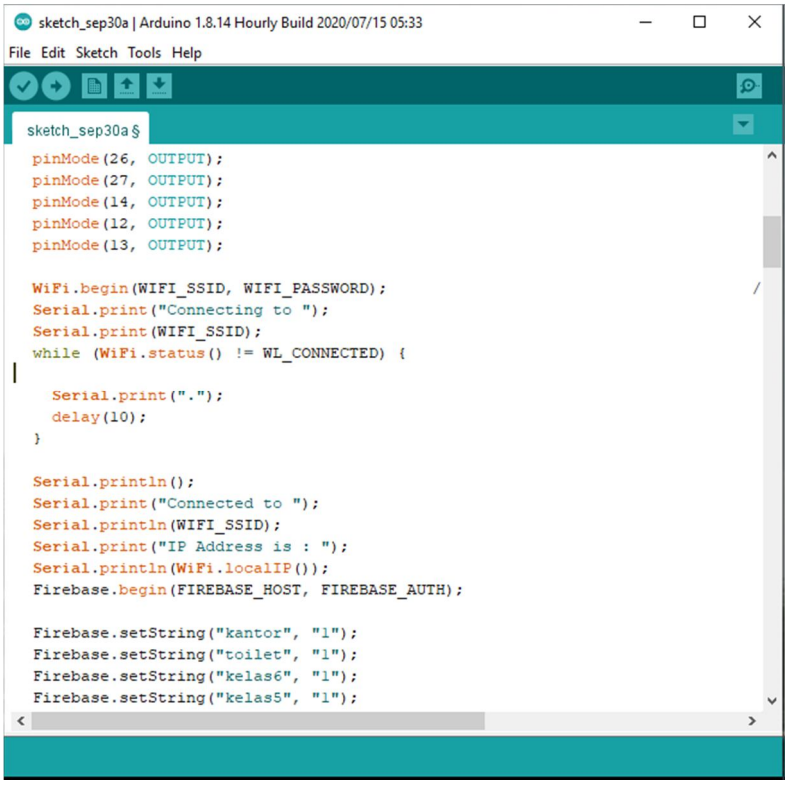


Gambar 4.10 Bentuk luar *Box*

L. Rancangan Perangkat Lunak (*Software*)

Setelah melakukan rancangan perangkat keras selanjutnya adalah melakukan perancangan perangkat lunak. Pada bagian perangkat lunak ini merupakan bagian pembuatan program yang akan di *upload* ke dalam *mikrokontroler* yaitu berupa perintah untuk mengelola data untuk diaplikasikan. Adapun bahasa pemrograman yang digunakan adalah berupa *sketch* program dalam bahasa C dan untuk membangun sistem kontrol lampu kita menggunakan *Mit App Inventor*.





```
sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33
File Edit Sketch Tools Help
sketch_sep30a §

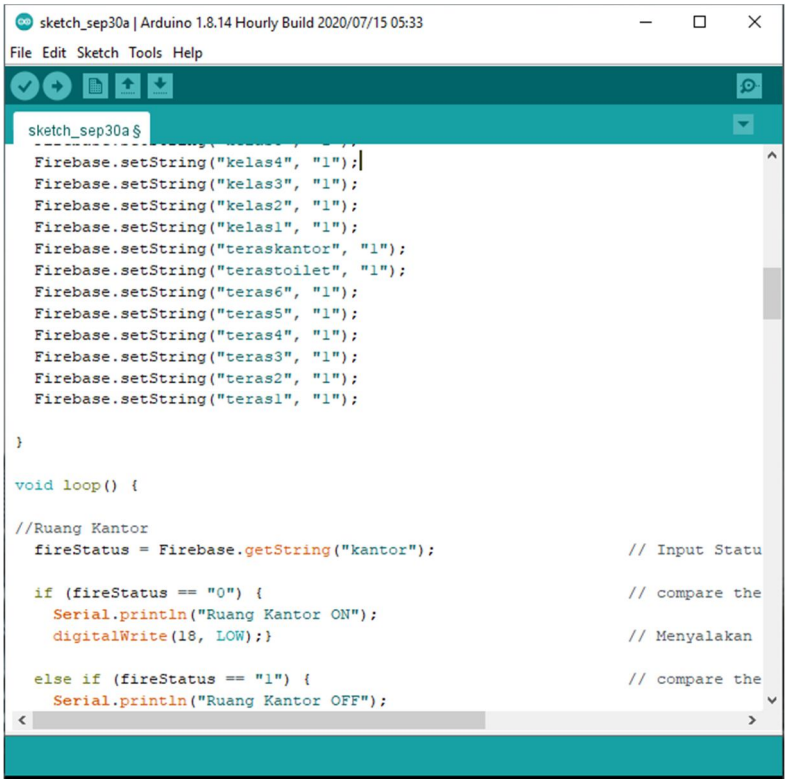
  pinMode(26, OUTPUT);
  pinMode(27, OUTPUT);
  pinMode(14, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);                                    /
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {

    Serial.print(".");
    delay(10);
  }

  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  Firebase.setString("kantor", "1");
  Firebase.setString("toilet", "1");
  Firebase.setString("kelas6", "1");
  Firebase.setString("kelas5", "1");
```



```
sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33
File Edit Sketch Tools Help
sketch_sep30a §

  Firebase.setString("kelas4", "1");
  Firebase.setString("kelas3", "1");
  Firebase.setString("kelas2", "1");
  Firebase.setString("kelas1", "1");
  Firebase.setString("teraskantor", "1");
  Firebase.setString("terastoilet", "1");
  Firebase.setString("teras6", "1");
  Firebase.setString("teras5", "1");
  Firebase.setString("teras4", "1");
  Firebase.setString("teras3", "1");
  Firebase.setString("teras2", "1");
  Firebase.setString("teras1", "1");

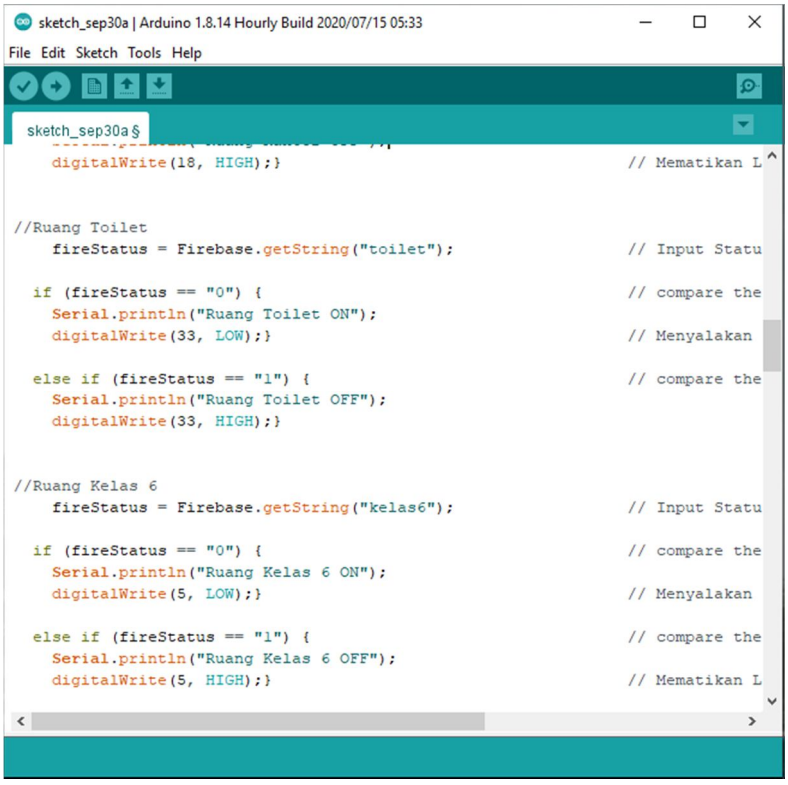
}

void loop() {

//Ruang Kantor
  fireStatus = Firebase.getString("kantor");                  // Input Statu

  if (fireStatus == "0") {                                    // compare the
    Serial.println("Ruang Kantor ON");
    digitalWrite(18, LOW);}                                   // Menyalakan

  else if (fireStatus == "1") {                               // compare the
    Serial.println("Ruang Kantor OFF");
```



sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33

File Edit Sketch Tools Help

sketch_sep30a §

```
    digitalWrite(18, HIGH);}                                    // Mematikan L


//Ruang Toilet
    fireStatus = Firebase.getString("toilet");                  // Input Statu

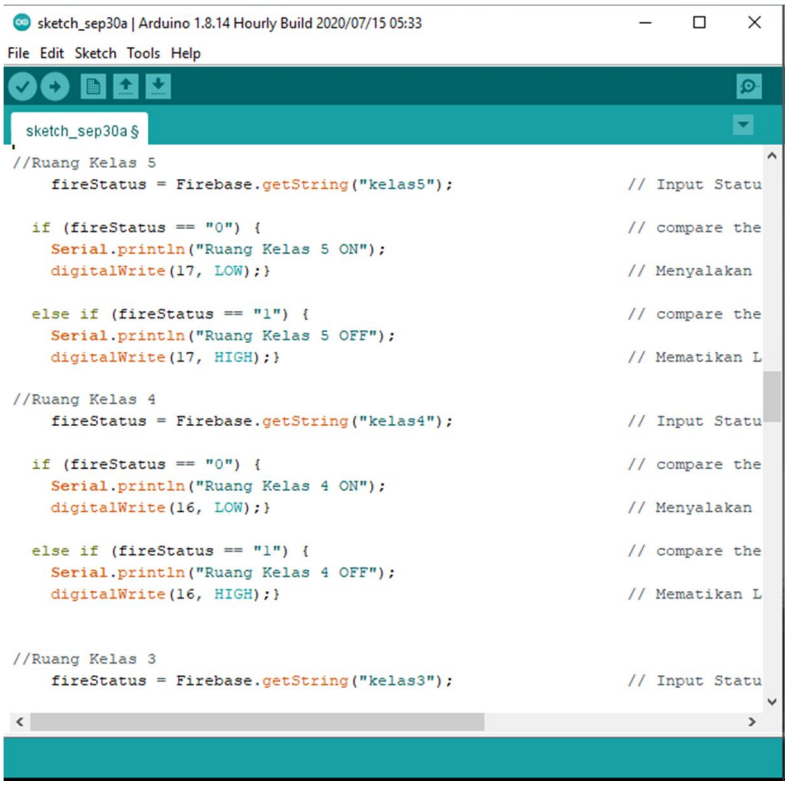
  if (fireStatus == "0") {                                      // compare the
    Serial.println("Ruang Toilet ON");
    digitalWrite(33, LOW);}                                     // Menyalakan

  else if (fireStatus == "1") {                                 // compare the
    Serial.println("Ruang Toilet OFF");
    digitalWrite(33, HIGH);}


//Ruang Kelas 6
    fireStatus = Firebase.getString("kelas6");                  // Input Statu

  if (fireStatus == "0") {                                      // compare the
    Serial.println("Ruang Kelas 6 ON");
    digitalWrite(5, LOW);}                                      // Menyalakan

  else if (fireStatus == "1") {                                 // compare the
    Serial.println("Ruang Kelas 6 OFF");
    digitalWrite(5, HIGH);}                                     // Mematikan L
```



sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33

File Edit Sketch Tools Help

sketch_sep30a §

```
//Ruang Kelas 5
    fireStatus = Firebase.getString("kelas5");                  // Input Statu

  if (fireStatus == "0") {                                      // compare the
    Serial.println("Ruang Kelas 5 ON");
    digitalWrite(17, LOW);}                                     // Menyalakan

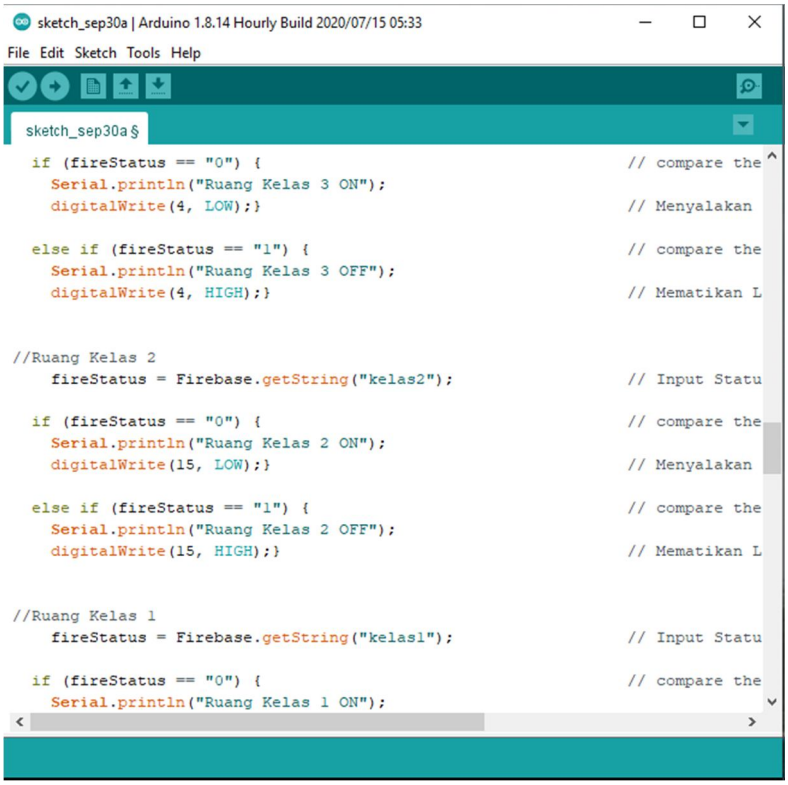
  else if (fireStatus == "1") {                                 // compare the
    Serial.println("Ruang Kelas 5 OFF");
    digitalWrite(17, HIGH);}                                    // Mematikan L

//Ruang Kelas 4
    fireStatus = Firebase.getString("kelas4");                  // Input Statu

  if (fireStatus == "0") {                                      // compare the
    Serial.println("Ruang Kelas 4 ON");
    digitalWrite(16, LOW);}                                     // Menyalakan

  else if (fireStatus == "1") {                                 // compare the
    Serial.println("Ruang Kelas 4 OFF");
    digitalWrite(16, HIGH);}                                    // Mematikan L


//Ruang Kelas 3
    fireStatus = Firebase.getString("kelas3");                  // Input Statu
```



sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33

File Edit Sketch Tools Help

sketch_sep30a §

```
  if (fireStatus == "0") {                                  // compare the
    Serial.println("Ruang Kelas 3 ON");
    digitalWrite(4, LOW);}                                  // Menyalakan

  else if (fireStatus == "1") {                             // compare the
    Serial.println("Ruang Kelas 3 OFF");
    digitalWrite(4, HIGH);}                                 // Mematikan L

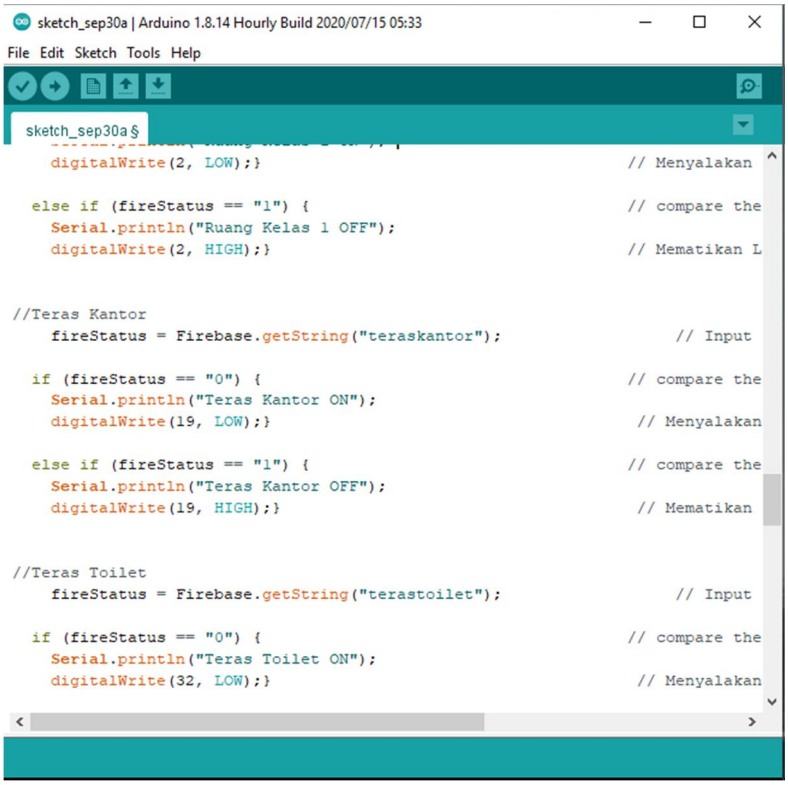

//Ruang Kelas 2
    fireStatus = Firebase.getString("kelas2");              // Input Statu

  if (fireStatus == "0") {                                  // compare the
    Serial.println("Ruang Kelas 2 ON");
    digitalWrite(15, LOW);}                                 // Menyalakan

  else if (fireStatus == "1") {                             // compare the
    Serial.println("Ruang Kelas 2 OFF");
    digitalWrite(15, HIGH);}                                // Mematikan L


//Ruang Kelas 1
    fireStatus = Firebase.getString("kelas1");              // Input Statu

  if (fireStatus == "0") {                                  // compare the
    Serial.println("Ruang Kelas 1 ON");
```



sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33

File Edit Sketch Tools Help

sketch_sep30a §

```
    digitalWrite(2, LOW);}                                  // Menyalakan

  else if (fireStatus == "1") {                             // compare the
    Serial.println("Ruang Kelas 1 OFF");
    digitalWrite(2, HIGH);}                                 // Mematikan L

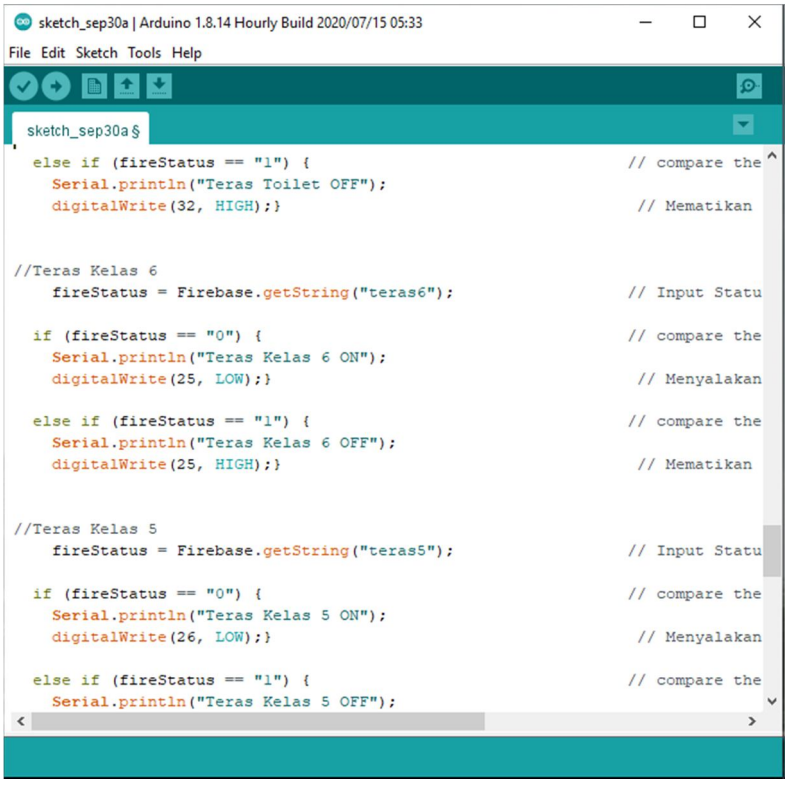

//Teras Kantor
    fireStatus = Firebase.getString("teraskantor");             // Input

  if (fireStatus == "0") {                                  // compare the
    Serial.println("Teras Kantor ON");
    digitalWrite(19, LOW);}                                  // Menyalakan

  else if (fireStatus == "1") {                             // compare the
    Serial.println("Teras Kantor OFF");
    digitalWrite(19, HIGH);}                                 // Mematikan


//Teras Toilet
    fireStatus = Firebase.getString("terastoilet");             // Input

  if (fireStatus == "0") {                                  // compare the
    Serial.println("Teras Toilet ON");
    digitalWrite(32, LOW);}                                  // Menyalakan
```



```
sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33
File  Edit  Sketch  Tools  Help
sketch_sep30a §

  else if (fireStatus == "1") {                          // compare the
    Serial.println("Teras Toilet OFF");
    digitalWrite(32, HIGH);}                              // Mematikan


//Teras Kelas 6
    fireStatus = Firebase.getString("teras6");           // Input Statu

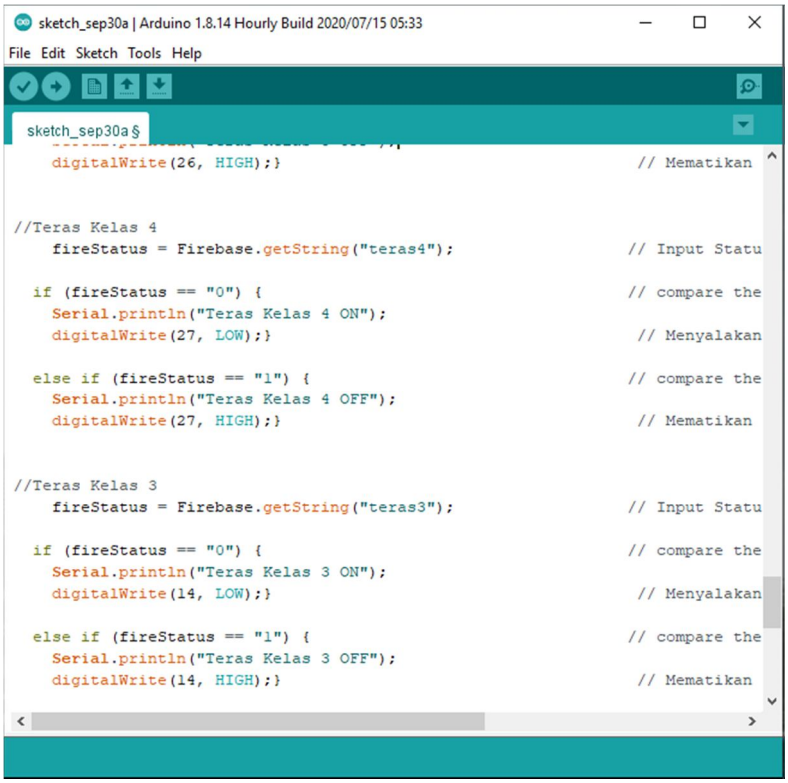
  if (fireStatus == "0") {                               // compare the
    Serial.println("Teras Kelas 6 ON");
    digitalWrite(25, LOW);}                               // Menyalakan

  else if (fireStatus == "1") {                          // compare the
    Serial.println("Teras Kelas 6 OFF");
    digitalWrite(25, HIGH);}                              // Mematikan


//Teras Kelas 5
    fireStatus = Firebase.getString("teras5");           // Input Statu

  if (fireStatus == "0") {                               // compare the
    Serial.println("Teras Kelas 5 ON");
    digitalWrite(26, LOW);}                               // Menyalakan

  else if (fireStatus == "1") {                          // compare the
    Serial.println("Teras Kelas 5 OFF");
```



```
sketch_sep30a | Arduino 1.8.14 Hourly Build 2020/07/15 05:33
File  Edit  Sketch  Tools  Help
sketch_sep30a §

    digitalWrite(26, HIGH);}                              // Mematikan


//Teras Kelas 4
    fireStatus = Firebase.getString("teras4");           // Input Statu

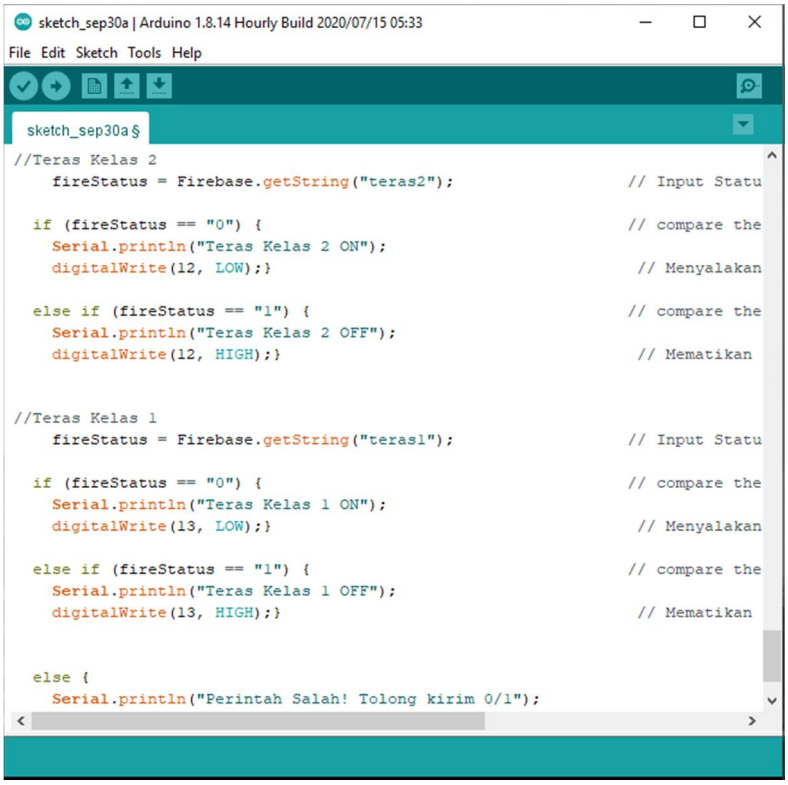
  if (fireStatus == "0") {                               // compare the
    Serial.println("Teras Kelas 4 ON");
    digitalWrite(27, LOW);}                               // Menyalakan

  else if (fireStatus == "1") {                          // compare the
    Serial.println("Teras Kelas 4 OFF");
    digitalWrite(27, HIGH);}                              // Mematikan


//Teras Kelas 3
    fireStatus = Firebase.getString("teras3");           // Input Statu

  if (fireStatus == "0") {                               // compare the
    Serial.println("Teras Kelas 3 ON");
    digitalWrite(14, LOW);}                               // Menyalakan

  else if (fireStatus == "1") {                          // compare the
    Serial.println("Teras Kelas 3 OFF");
    digitalWrite(14, HIGH);}                              // Mematikan
```

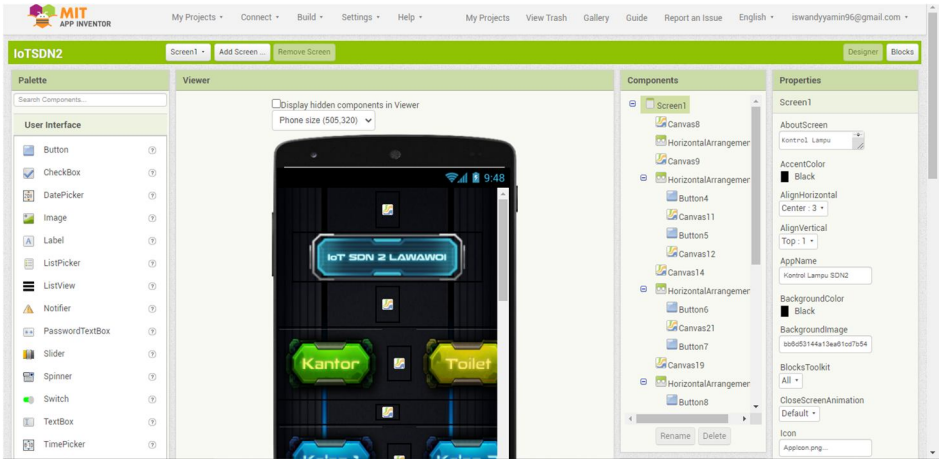


Gambar 4.11 *Sketch Arduino Ide* (bahasa C)

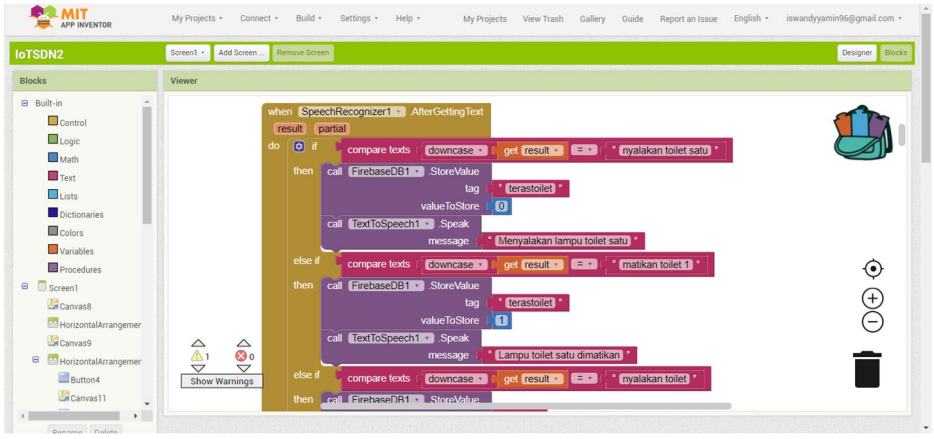
Pada gambar *arduino ide* (*sketch*) sebuah *software* untuk memprogram *arduino* dan komponen pendukung lainnya dengan menggunakan *sintaks* pemrograman bahasa C. *Arduino ide* dibuat dari bahasa pemrograman *Java* yang dilengkapi dengan *library C/C++* yang membuat operasi *input/output* lebih mudah.

M. Rancangan Aplikasi *Android*

Aplikasi android dibuat dengan menggunakan *Mit App Inventor.* Aplikasi ini bertujuan untuk mengontrol *Prototype* Sistem kontrrol lampu SDN 2 Lawawoi dari jarak jauh dengan memanfaatkan koneksi jaringan internet dan *Android.*



Gambar 4.12 Tampilan *Screen Project*



Gambar 4.13 Tampilan *block Screen*

Pada gambar diatas merupakan pembuatan apalikasi *Android APK* pada aplikasi *Mit App Inventor.* Dengan *app inventor*, pengguna bisa melakukan pemrograman komputer untuk menciptakan aplikasi perangkat lunak dengan sistem operasi berbasi *android. App inventor* ini berbasis *visual block* programming karena memungkinkan pengguna bisa menggunakan, melihat, menyusun dan men-*drag dan drops block* yang merupakan simbol perintah dan

fungsi *event handler* untuk menciptakan sebuah aplikasi yang bisa berjalan di sistem *android.*

1. Tampilan menu utama

   Menu utama terdiri dari ruangan kantor, toilet, kelas 1, kelas 2, kelas 3, kelas 4, kelas 5, dan kelas 6. Selain itu juga terdapat tombol "Ucapkan Perintah" yang berfungsi untuk mengontrol lampu hanya dengan mengucapkan perintah, seperti: "Nyalakan Lampu Kelas 1", "Lampu Kelas 1 Menyala", "Kelas 1 *ON*" dan untuk mematikan lampu perintahnya adalah: "Matikan Lampu Kelas 1", "Lampu Kelas 1 Mati", "Kelas 1 *OFF*" perintah setiap ruangan hampir sama, yang membedakan hanya nama kelas atau ruangan yang ingin dikontrol lampunya.

   Tombol "Semua Lampu" berfungsi untuk mengontrol semua lampu sekaligus. Jika ditekan sekali maka lampu akan menyala, dan tombol akan berubah warna menjadi hijau dengan tulisan *"ON ALL"* yang berarti kondisi semua lampu sendang menyala. Jika ditekan lagi maka tombol akan berubah menjadi merah dengan tulisan *"OFF ALL"* yang menandakan kondisi semua lampu sedang mati.

   Pada bagian paling bawah terdapat opsi untuk mengatur waktu menyala dan mati lampu. Jadi dengan ini kita dapat menjadwalkan kapan lampu menyala dan kapan lampu akan mati secara otomatis. Ketika jadwal diaktifkan maka tombol akan berwarna merah dengan tulisan

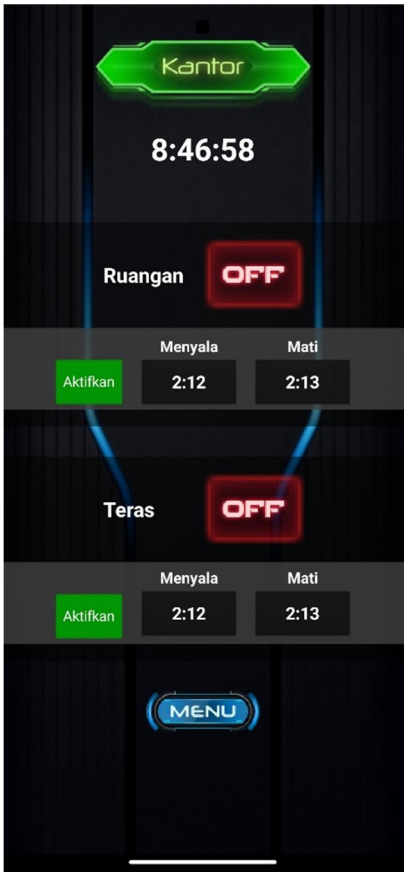
"Matikan", sedangkan saat jadwal tidak diaktifkan maka tombol akan berwarna hijau dengan tulisan "Aktifkan".



Gambar 4.14 Tampilan menu utama

2. Tampilan ruangan kantor

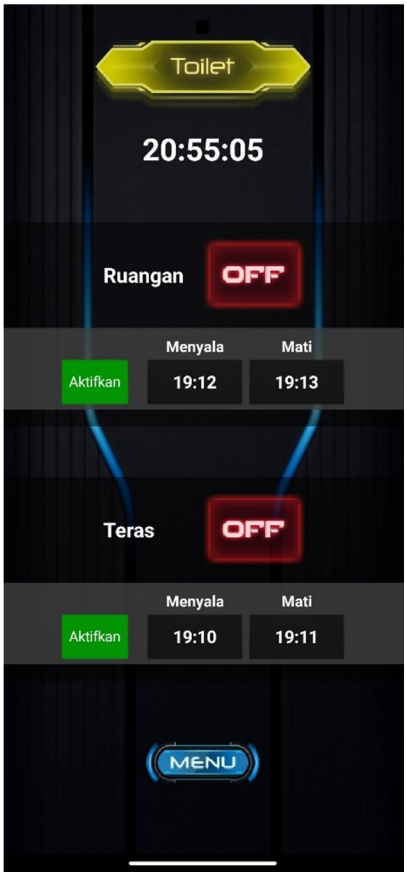
Tampilan ruangan kantor terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol "Menu" untuk kembali ke menu utama.



Gambar 4.15 Tampilan ruangan kantor

3. Tampilan ruangan toilet

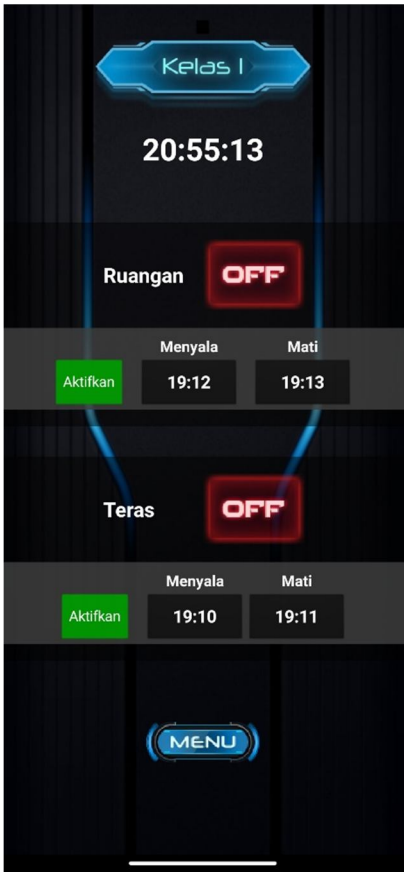
Tampilan ruangan toilet terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.16 Tampilan ruangan toilet

4. Tampilan ruangan kelas 1

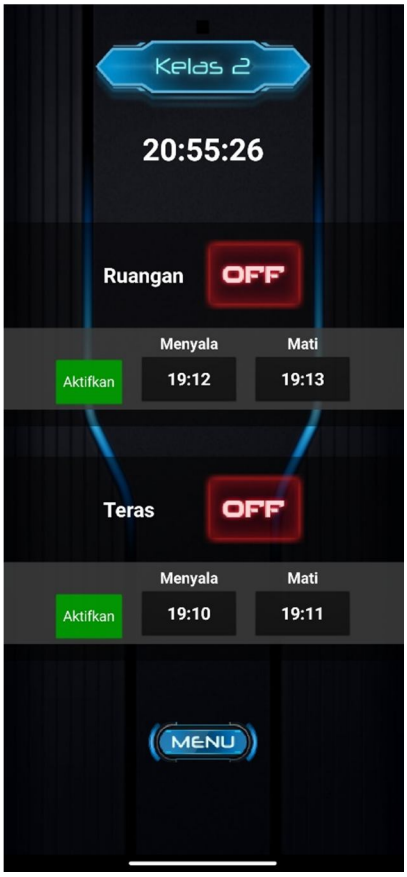
Tampilan ruangan kelas 1 terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.17 Tampilan ruangan kelas 1

5. Tampilan ruangan kelas 2

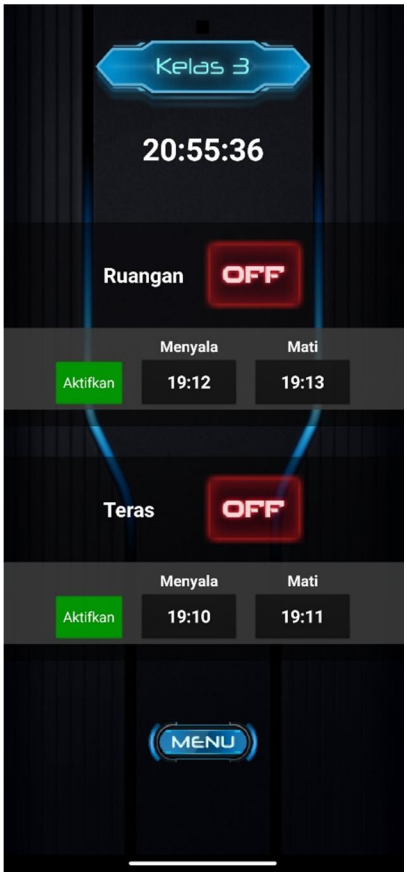
Tampilan ruangan kelas 2 terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.18 Tampilan ruangan kelas 2

6. Tampilan ruangan kelas 3

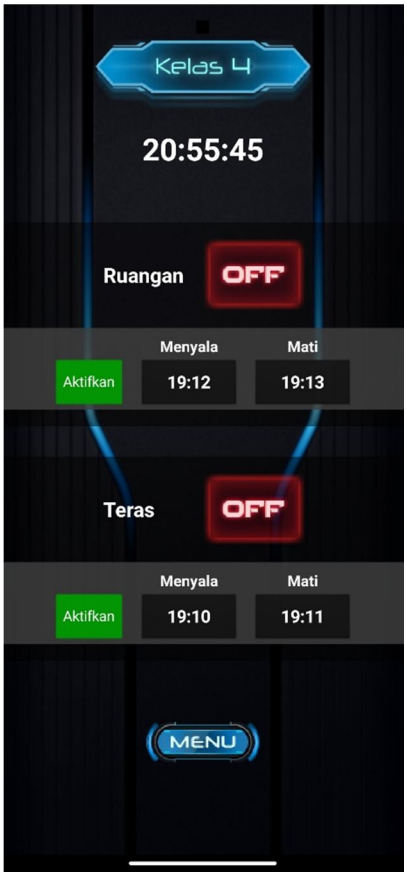
Tampilan ruangan kelas 3 terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.19 Tampilan ruangan kelas 3

7. Tampilan ruangan kelas 4

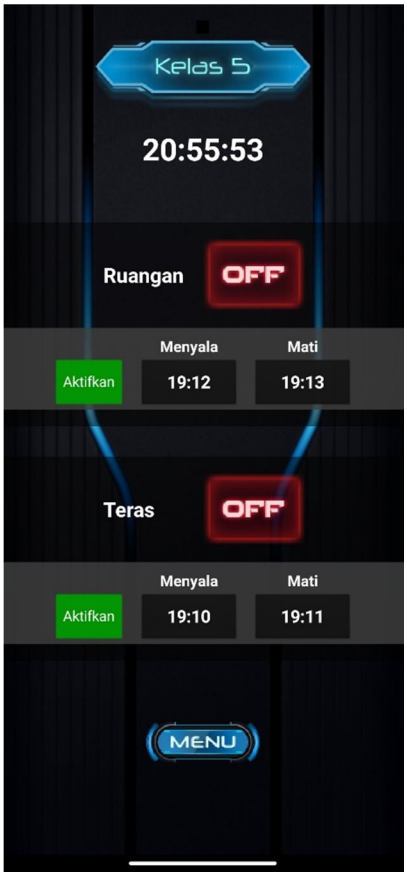
Tampilan ruangan kelas 4 terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.20 Tampilan ruangan kelas 4

8. Tampilan ruangan kelas 5

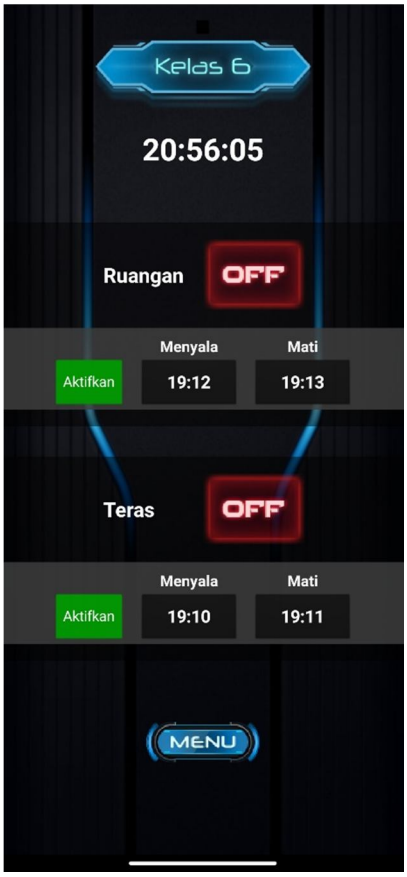
Tampilan ruangan kelas 5 terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.21 Tampilan ruangan kelas 5

9. Tampilan ruangan kelas 6

Tampilan ruangan kelas 6 terdiri dari jam atau waktu di bagian atas secara *realtime*. Kemudian terdapat tombol untuk menyalakan lampu ruangan atau teras, serta opsi untuk mengatur jadwal kapan lampu akan menyala dan kapan lampu akan mati secara otomatis. Dan tombol “Menu” untuk kembali ke menu utama.



Gambar 4.22 Tampilan ruangan kelas 6

N. PEMBAHASAN

1. Metode Pengujian

   Metode pengujian sistem merupakan proses eksekusi suatu program atau sistem keseluruhan. Pengujian menunjuka bahwa fungsi perangkat keras dan perangkat lunak bekerja sesuai dengan spesifikasi bahwa persyaratan kinerja telah terpenuhi.

2. Pengujian *Black Box*

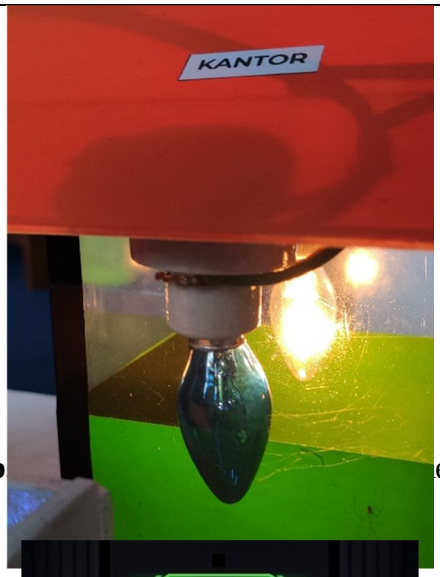
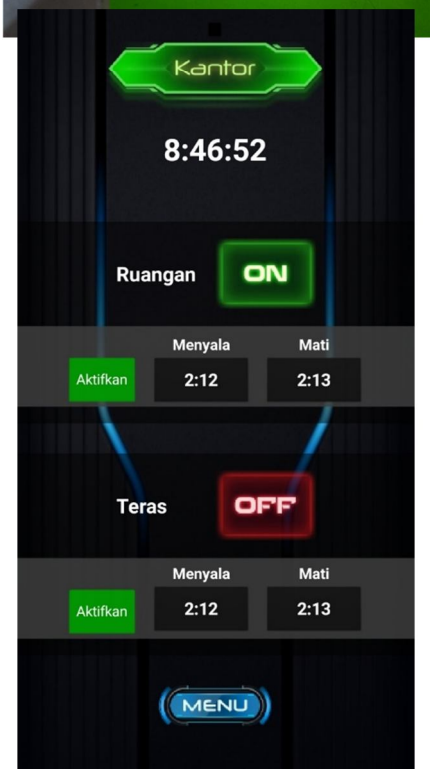
Pengujian *Black-box* merupakan suatu pengujian pada *interface* perangkat lunak. Pengujian *Black-box* memungkinan prekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Perlu juga diketahui bahwa pengujian *Black-box* juga memperhatikan struktur kontrol. Meskipun didesain sedemikian rupa untuk mengungkap kesalahan. Pengujian *Black-box* digunakan untuk memperlihatkan bahwa fungsi - fungsi perangkat lunak adalah operasional bahwa input diterima dengan baik dan output dihasilkan dengan tepat serta integritas informasi *external* (misalnya file data).

Pengujian *Black-Box* menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur logika internal perangkat lunak tersebut.

a. Pengujian Tahap 1

Tabel 4.6. Pengujian *Prototype* ruangan kantor

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot
 **Gamb** enyala  **Gambar 4.24** Tampilan aplikasi saat lampu kantor menyala

b. Pengujian Tahap 2

Tabel 4.7. Pengujian *Prototype* teras kantor

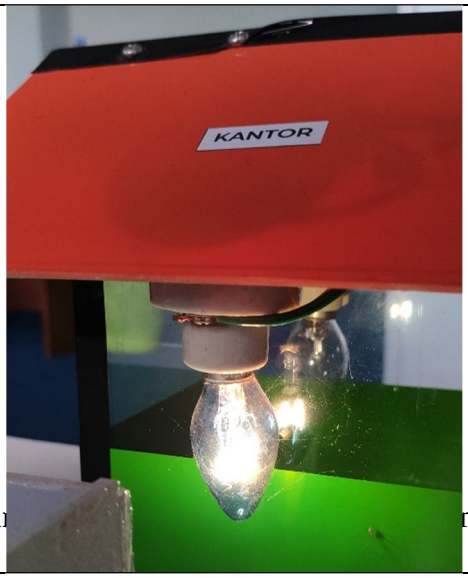
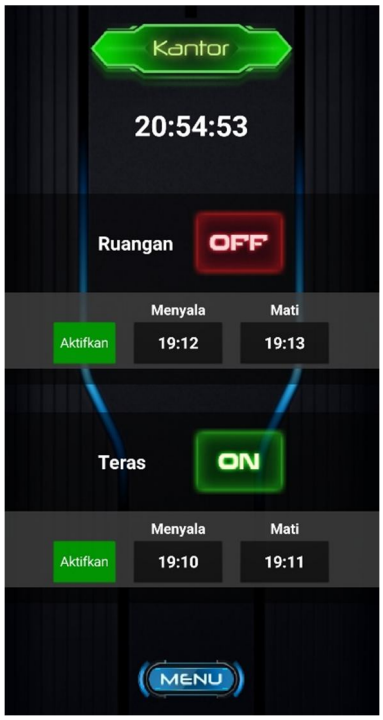
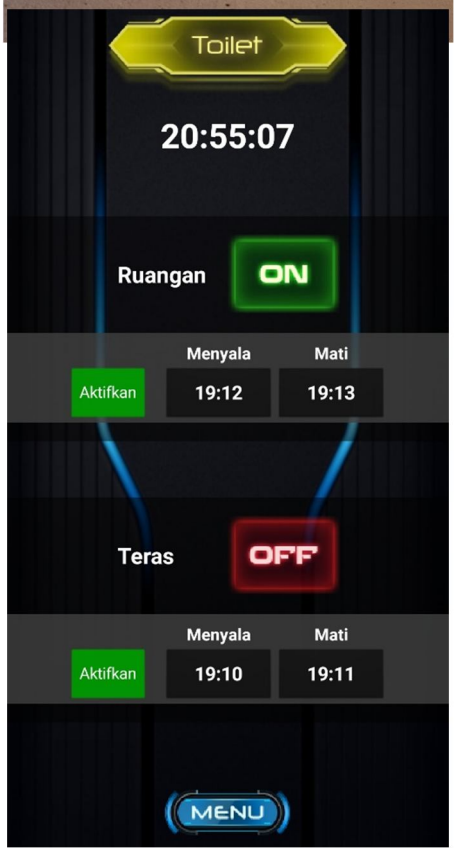
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot
 **Gambar 4.26** Tampilan aplikasi saat lampu teras kantor menyala

c. Pengujian Tahap 3

Tabel 4.8. Pengujian *Prototype* toilet

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot
 

Gam nyala

Gambar 4.28 Tampilan aplikasi saat lampu toilet menyala

d. Pengujian Tahap 4

Tabel 4.9. Pengujian *Prototype* teras toilet

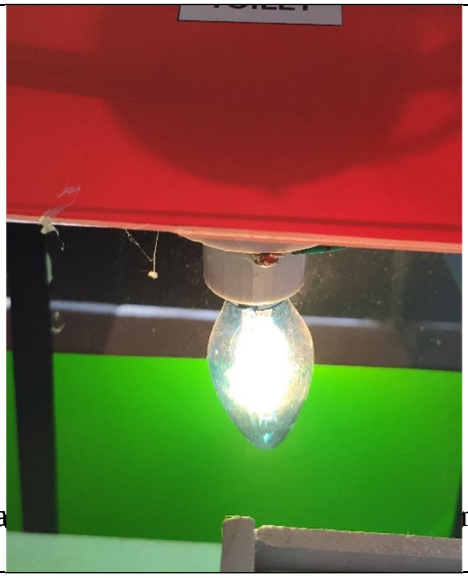
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
 **Gamba** menyala		

Foto dan Screenshot
 **Gambar 4.30** Tampilan aplikasi saat lampu teras toilet menyala

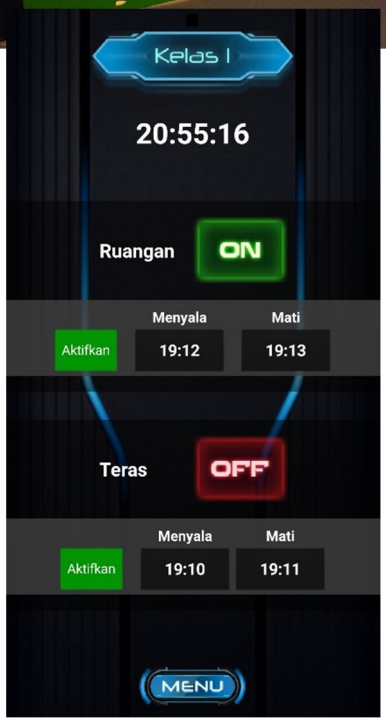
e. Pengujian Tahap 5

Tabel 4.10. Pengujian *Prototype* kelas 1

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot


Gam nyala



Gambar 4.32 Tampilan aplikasi saat lampu kelas 1 menyala

f. Pengujian Tahap 6

Tabel 4.11. Pengujian *Prototype* teras kelas 1

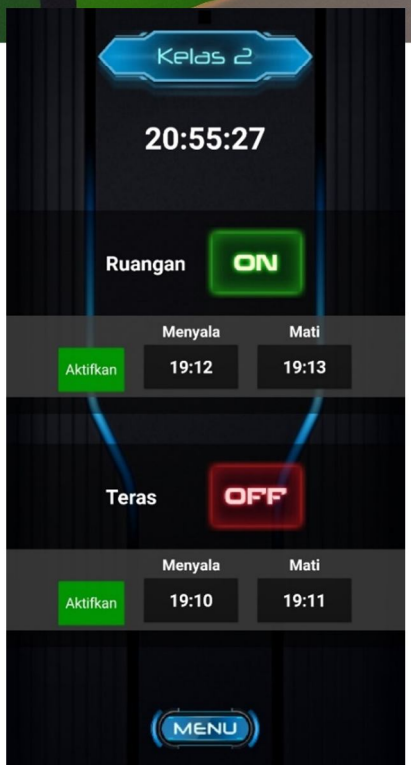
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot
 **Gambar 4.34** Tampilan aplikasi saat lampu teras kelas 1 menyala

g. Pengujian Tahap 7

Tabel 4.12. Pengujian *Prototype* kelas 2

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot
 **Gamb** enyala  **Gambar 4.36** Tampilan aplikasi saat lampu kelas 2 menyala

h. Pengujian Tahap 8

Tabel 4.13. Pengujian *Prototype* teras kelas 2

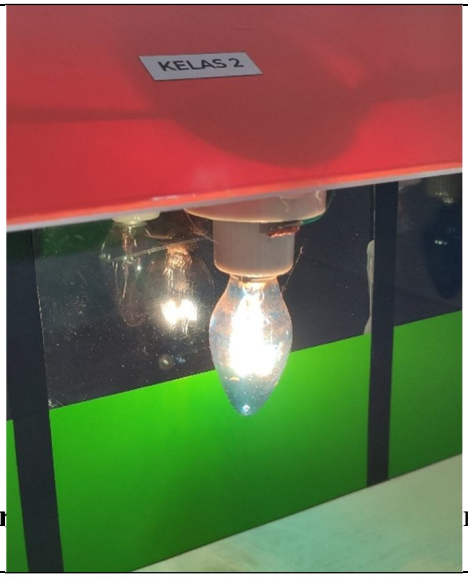
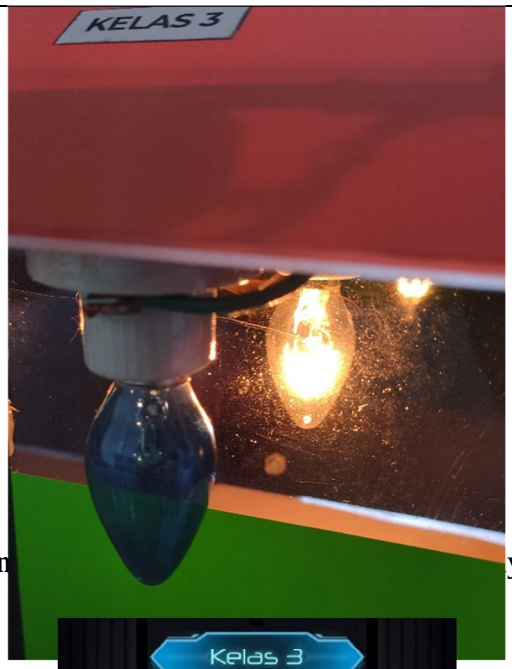
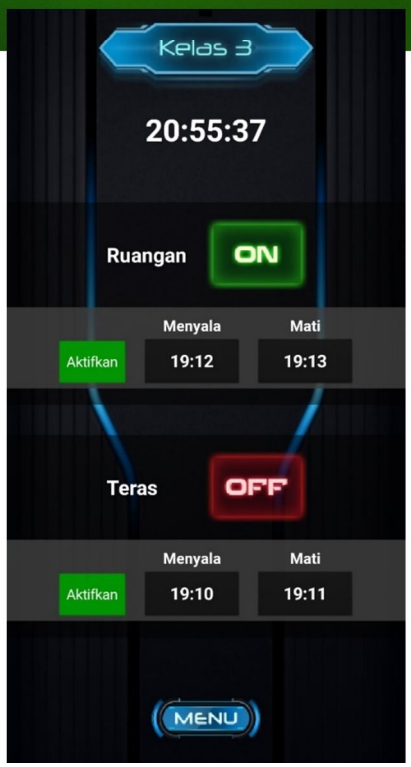
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot
 **Gambar 4.38** Tampilan aplikasi saat lampu teras kelas 2 menyala

i. Pengujian Tahap 9

Tabel 4.14. Pengujian *Prototype* kelas 3

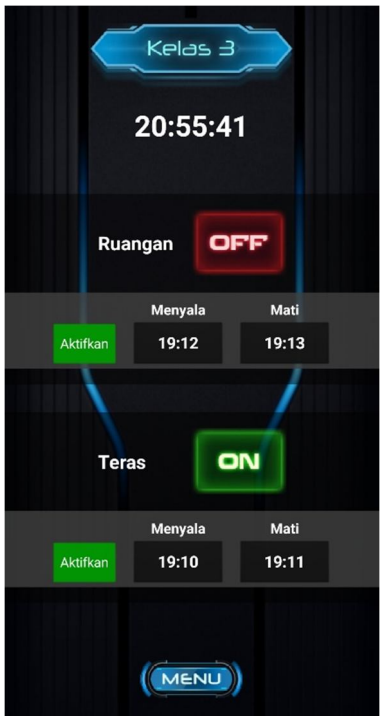
Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot
 **Gam** yala  **Gambar 4.40** Tampilan aplikasi saat lampu kelas 3 menyala

j. Pengujian Tahap 10

Tabel 4.15. Pengujian *Prototype* teras kelas 3

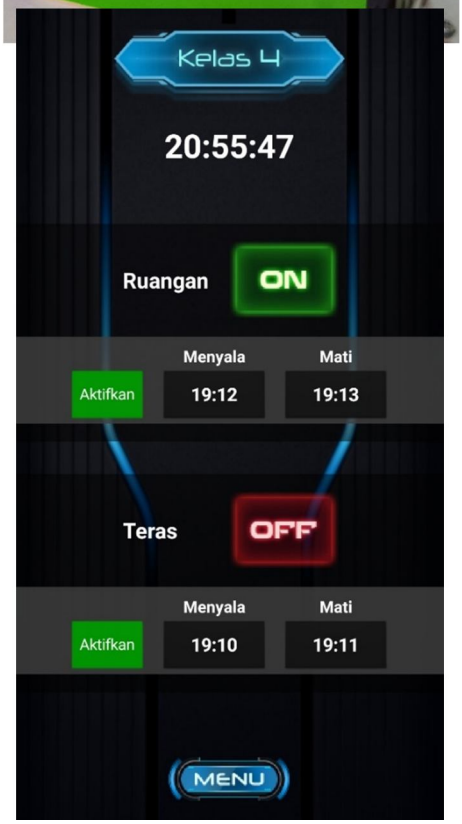
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot
 **Gambar 4.42** Tampilan aplikasi saat lampu teras kelas 3 menyala

k. Pengujian Tahap 11

Tabel 4.16. Pengujian *Prototype* kelas 4

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot
 **Gamb** enyala  **Gambar 4.44** Tampilan aplikasi saat lampu kelas 4 menyala

l. Pengujian Tahap 12

Tabel 4.17. Pengujian *Prototype* teras kelas 4

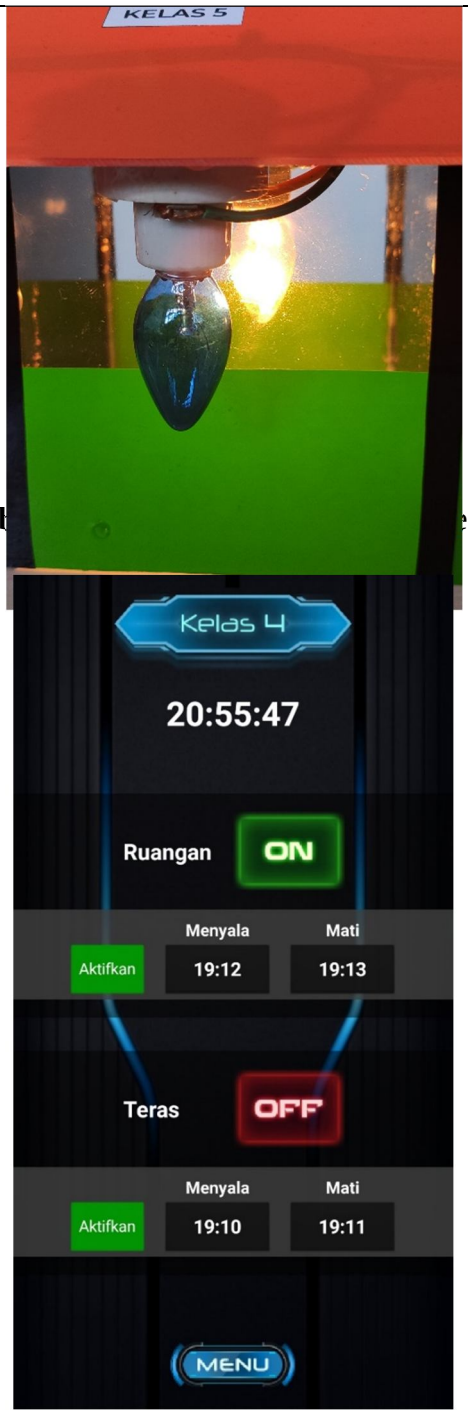
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot
 **Gambar 4.46** Tampilan aplikasi saat lampu teras kelas 4 menyala

m. Pengujian Tahap 13

Tabel 4.18. Pengujian *Prototype* kelas 5

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot


Gamb **enyala**

Gambar 4.48 Tampilan aplikasi saat lampu kelas 5 menyala

n. Pengujian Tahap 14

Tabel 4.19. Pengujian *Prototype* teras kelas 5

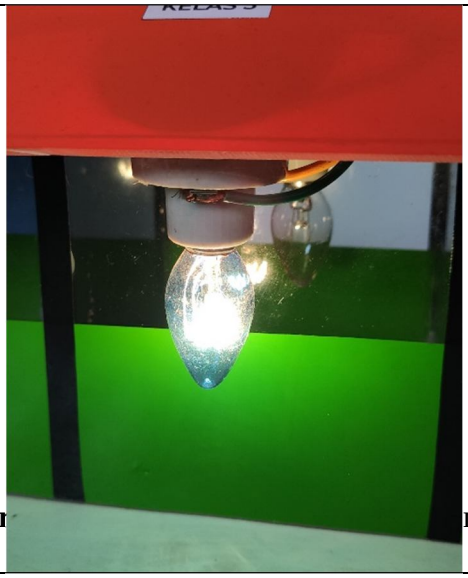
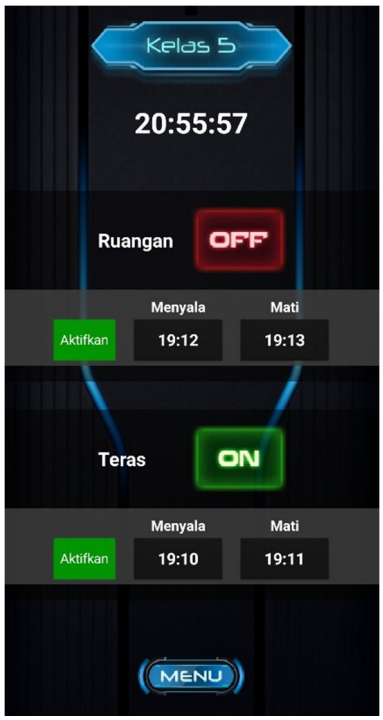
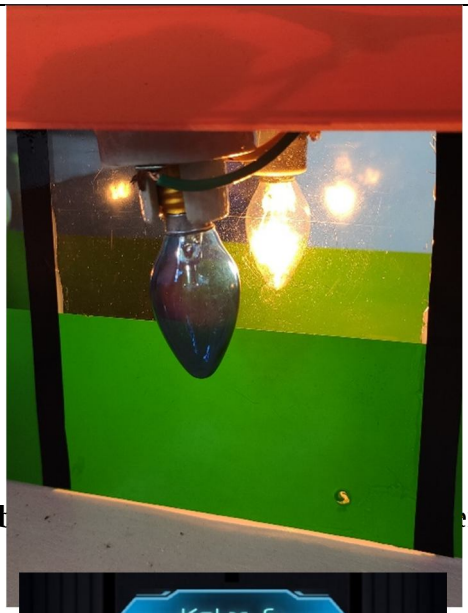
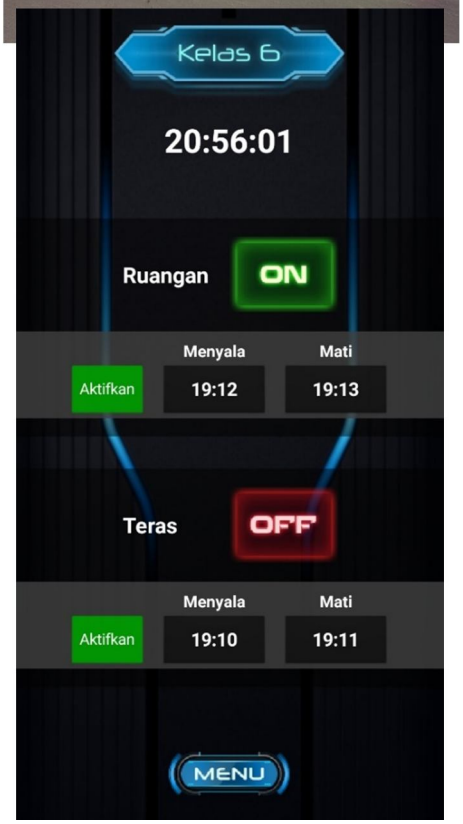
Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot
 **Gambar 4.50** Tampilan aplikasi saat lampu teras kelas 5 menyala

o. Pengujian Tahap 15

Tabel 4.20. Pengujian *Prototype* kelas 6

Uji coba	Hasil	Keterangan
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.

Foto dan Screenshot

Gamb nyala

Gambar 4.52 Tampilan aplikasi saat lampu teras kelas 6 menyala

p. Pengujian Tahap 16

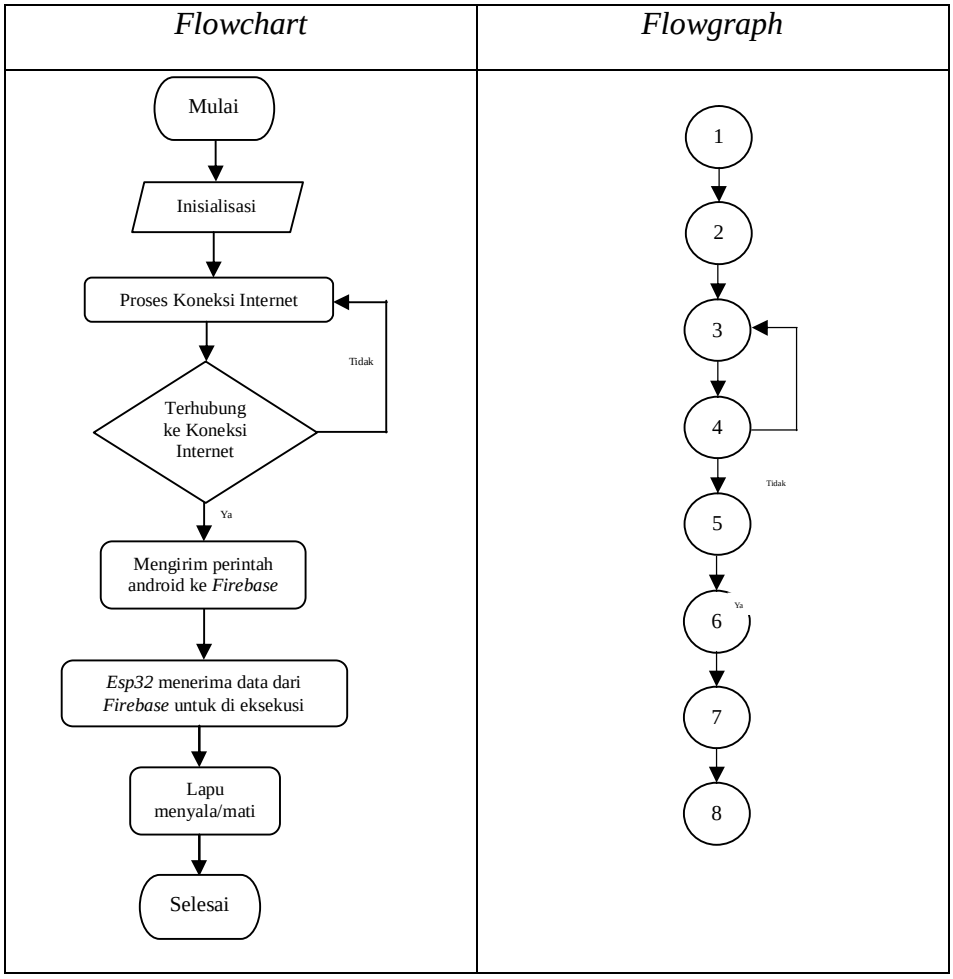
Tabel 4.21. Pengujian *Prototype* teras kelas 6

Uji coba	**Hasil**	**Keterangan**
Jika petugas menekan *button ON* pada aplikasi *android*, maka lampu akan menyala sesuai dengan nama *button* yang ditekan.	Lampu Menyala	Karena *Esp32* menerima data yang sudah ditentukan sehingga mengaktikan *relay* agar lampu menyala dan status *button* pada aplikasi *android* menjadi *ON*.
Foto dan Screenshot		
Gambar  menyala		

Foto dan Screenshot


Gambar 4.54 Tampilan aplikasi saat lampu teras kelas 6 menyala

3. Pengujian White Box



Gambar 4.55 *Flowchart* dan *Flowgrap* Kontrol Lampu

Dari gambar *Flowgraph* Irigasi Pembagian di atas dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung C*yclomatic Complexcity* V(G) dari *Egde* dan *Node*:

DenganRumus : V (G) = E – N + 2

Dengan E (*edge*) = 8

N (*Node*) = 8

P (Predikat *Node*) = 1

Penyelesaian : V(G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat *Node* (P) = P + 1

= 1 + 1

= 2

2) Berdasarkan perhitungan C*yclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

3) *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 4 – 3

Path 2 = 1 – 2 – 3 – 4 – 5– 6 – 7 – 8

4) *Graph Matrix*

5)

	1	2	3	4	5	6	7	8	Koneksi
1		1							1-1= 0
2			1						1-1= 0
3				1					1-1= 0
4			1		1				2-1= 1
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									

Cyclomatic Complexity = 1+1 =2

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian dan hasil pengujian peracangan sistem kontrol lampu berbasis *internet of things* dan *android* pada SDN 2 Lawawoi dapat ditarik kesimpulan bahwa :

1. Sistem yang dibuat dapat dikontrol serta mampu berjalan sebagaimana mestinya tanpa terjadi kesalahan sehingga mendapatkan hasil; Jika tombol "*ON*" ruang kantor ditekan pada aplikasi, maka lampu ruang kantor pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang kantor ditekan pada aplikasi, maka lampu ruang kantor pada *prototype* akan mati. Dan Jika tombol "*ON*" ruang toilet ditekan pada aplikasi, maka lampu ruang toilet pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang toilet ditekan pada aplikasi, maka lampu ruang toilet pada *prototype* akan mati. Dan Jika tombol "*ON*" ruang kelas 1 ditekan pada aplikasi, maka lampu ruang kelas 1 pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang kelas 1 ditekan pada aplikasi, maka lampu ruang kelas 1 pada *prototype* akan mati. Dan Jika tombol "*ON*" ruang kelas 2 ditekan pada aplikasi, maka lampu ruang kelas 2 pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang kelas 2 ditekan pada aplikasi, maka lampu ruang kelas 2 pada *prototype* akan mati. Jika tombol "*ON*" ruang kelas 3 ditekan pada aplikasi, maka lampu ruang kelas 3 pada *prototype* akan menyala. Dan

jika tombol "*OFF*" ruang kelas 3 ditekan pada aplikasi, maka lampu ruang kelas 3 pada *prototype* akan mati. Dan Jika tombol "*ON*" ruang kelas 4 ditekan pada aplikasi, maka lampu ruang kelas 4 pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang kelas 4 ditekan pada aplikasi, maka lampu ruang kelas 4 pada *prototype* akan mati. Jika tombol "*ON*" ruang kelas 5 ditekan pada aplikasi, maka lampu ruang kelas 5 pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang kelas 5 ditekan pada aplikasi, maka lampu ruang kelas 5 pada *prototype* akan mati. Dan Jika tombol "*ON*" ruang kelas 6 ditekan pada aplikasi, maka lampu ruang kelas 6 pada *prototype* akan menyala. Dan jika tombol "*OFF*" ruang kelas 6 ditekan pada aplikasi, maka lampu ruang kelas 6 pada *prototype* akan mati.

2. Sistem kontrol lampu SDN 2 Lawawoi juga dapat di kontrol dengan mengatur jadwal menyala dan mati lampu. Yang mana jika kita mengatur waktu menyala lampu ruangan kantor pada aplikasi, maka lampu ruangan kantor pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kantor pada aplikasi, maka lampu ruangan kantor pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan toilet pada aplikasi, maka lampu ruangan toilet pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan toilet pada aplikasi, maka lampu ruangan toilet pada *prototype* akan mati sesuai dengan waktu

yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan kelas 1 pada aplikasi, maka lampu ruangan kelas 1 pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kelas 1 pada aplikasi, maka lampu ruangan kelas 1 pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan kelas 2 pada aplikasi, maka lampu ruangan kelas 2 pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kelas 2 pada aplikasi, maka lampu ruangan kelas 2 pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan kelas 3 pada aplikasi, maka lampu ruangan kelas 3 pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kelas 3 pada aplikasi, maka lampu ruangan kelas 3 pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan kelas 4 pada aplikasi, maka lampu ruangan kelas 4 pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kelas 4 pada aplikasi, maka lampu ruangan kelas 4 pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan kelas 5 pada aplikasi, maka lampu ruangan kelas 5 pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kelas 5

pada aplikasi, maka lampu ruangan kelas 5 pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu menyala lampu ruangan kelas 6 pada aplikasi, maka lampu ruangan kelas 6 pada *prototype* akan menyala sesuai dengan waktu yang telah ditentukan. Dan jika kita mengatur waktu mati lampu ruangan kelas 6 pada aplikasi, maka lampu ruangan kelas 6 pada *prototype* akan mati sesuai dengan waktu yang telah ditentukan.

3. Selain itu, lampu juga dapat dikontrol dengan menggunakan opsi perintah suara. Dengan kata lain, cukup dengan mengucapkan format perintah: “Nyakalan lampu kelas 1”, ”Lampu kelas 1 menyala” atau “Matikan lampu kelas 1”, “Lampu kelas 1 mati” maka lampu akan menyala atau mati berdasarkan perintah yang diucapkan. Untuk lampu yang lain, selebihnya hanya perlu mengganti perintah sesuai dengan nama kelas atau ruangan yang ingin dikontrol.

B. Saran

Berdasarkan hasil penelitian, maka penulis memberikan saran sebagai berikut:

1. Untuk pengembangan kedepannya diharapkan sistem yang dibuat tidak hanya dapat mengontrol lampu saja, namun juga bisa untuk mengontrol alat elektroknik seperti kipas angin, *TV* dan lainnya.

2. Diharapkan dalam pembuatan sistem kontrol lampu berbasis *internet of things* dan *android* pada SDN 2 Lawawoi dapat mengontrol lampu berdasarkan tanggal dan waktu yang telah ditentukan.
3. Untuk penelitian sejenis selanjutnya dapat menggunakan bahasa pemrogram yang lain terkhusus pada pembuatan aplikasi *Android APK*.

DAFTAR PUSTAKA

Andrianto & Arief Susanto, "APLIKASI PENGONTROL JARAK JAUH PADA LAMPU RUMAH BERBASIS *ANDROID*" Teknik Informatika, Fakultas Teknik," Universitas Muria Kudus, 2015.

Andri Susanto & Ismail Darisman Jauhari, "Rancang Bangun Aplikasi *Android* Untuk Kontrol Lampu Gedung Menggunakan Media *Bluetooth* Berbasis *Arduino Uno*," Teknik Elektro Universitas Muhammadiyah Tangerang 2018.

B. Hohensee, *Android for Beginners - Developing apps using Android Studio*. Sweden, 2015.

Djuandi, Feri., 2015. Pengenalan *Arduino*. Jakarta: Elexmedia.

F.Dwi, H.Kuswara, Kaharudin, "Alat Kendali Lampu Rumah Menggunakan *Bluetooth* Berbasis *Android*," Tek. Informatika STMIK PalComTech, 2015.

Kristanto, Andi., 2015. Jaringan Komputer. Yogyakarta: Graha Ilmu.

Karomah, R.F. Christianti, W.Eka, "Rancang Bangun Sistem Kendali Jarak Jauh *ON/OFF* Lampu dan *Air Conditioner* (AC) Berbasis *Arduino* Melalui Internet," DIII Sekolah Tinggi Teknologi Telematika Telkom, 2015.

Leo Ade Putra & Arif Rahman Hakim, "Sistem Kendali Lampu Cerdas Pada *Smarthome* Berbasis *Android* mengunakan Metode *Fuzzy Logic Control*," Universitas Putera Batam, 2018.

Mehta, M. (2015). *ESP 8266: A BREAKTHROUGH IN WIRELESS SENSOR NETWORKS AND INTERNET OF THINGS.*

Sumenge., 2015. Perancangan Kendali Lampu Berbasis *Android*. Jurnal Teknik Elektro dan Komputer. Vol 3, No. 2, April 2015.

Sutono., 2015. Perancangan Sistem Aplikasi Otomatisasi Lampu Penerangan Menggunakan Sensor Gerak Dan Sensor Cahaya Berbasis *Arduino Uno*. Jurnal UNIKOM. Vol 12, No. 2, Januari 2015.

Safaat, N. (2015). Pemrograman Amplikasi *Mobile Smartphone* dan *Tablet PC* Berbasis *Android*. Bandung: Penerbit Infromatika.

Wicaksono, H. (n.d.). Catatan Kuliah "Automasi 1". Teknik Elektro - Universitas Kristen Petra.