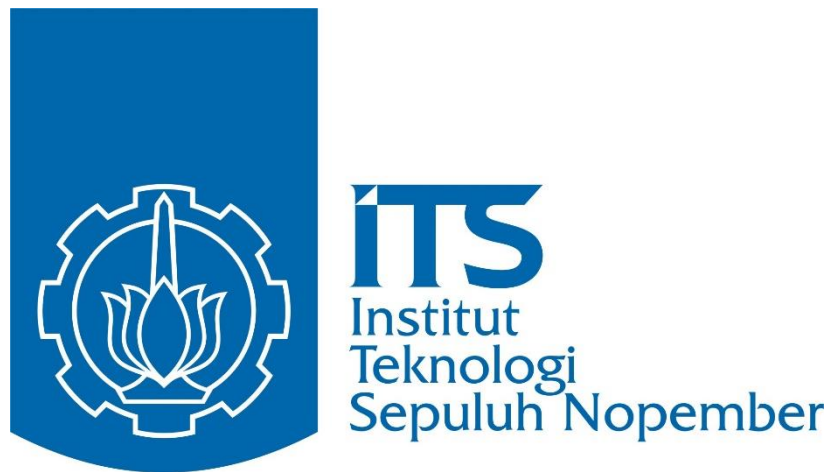


**LAPORAN AKHIR PRAKTIKUM
SISTEM ELEKTRONIKA TERTANAM**

**SMART HOME DEVICE AUTOMATION MENGGUNAKAN
RTC DS3231 DAN MODUL BLUETOOTH HC-05**



Disusun oleh:

Patrick Naga Kencana

NRP:

07111840000103

**PROGRAM STUDI ELEKTRONIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
2021**

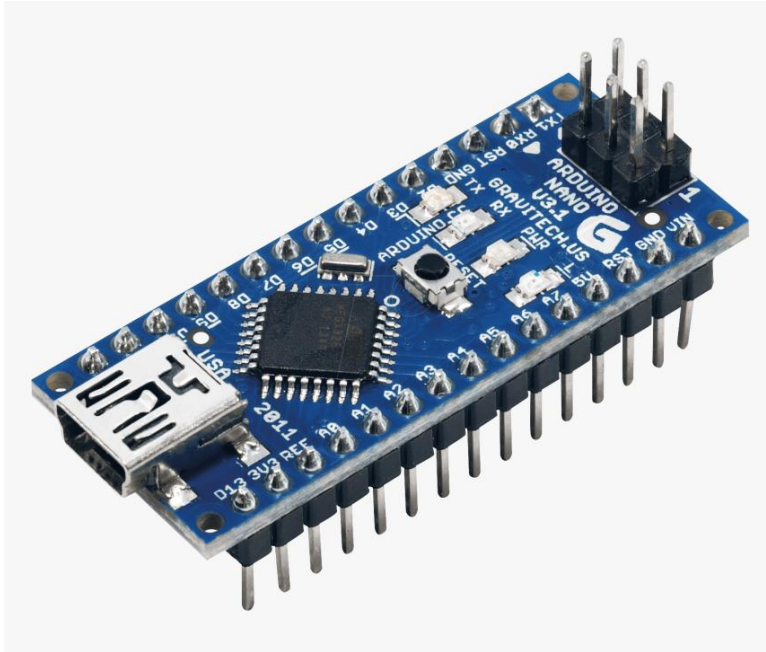
I

PENJELASAN ALAT

Rangkaian *Smart Home Device Automation* ini dibuat agar memiliki tujuan untuk membantu kita dan juga memudahkan kita dalam mengontrol peralatan listrik dalam rumah. Dibuatnya rangkaian ini diharapkan bisa membuat beberapa peralatan rumah bisa terkontrol otomatis, khususnya lampu depan rumah. Dengan itu, maka diharapkan rumah kita akan terlihat selalu terjaga karena lampu depan rumah akan selalu otomatis menyala / mati pada jam tertentu, walaupun penghuninya sedang tidak ada di rumah. Alat ini juga ditambahkan modul Bluetooth, sehingga kita bisa mengontrol lampu / peralatan rumah lainnya dengan alat yang sama. Pengontrolan dengan Bluetooth juga bisa dilakukan menggunakan Handphone, dengan cukup hanya menginstall aplikasi yang telah dibuat. Dengan ini, maka mudahnya alat ini memiliki dua buah sistem pengontrol kontrol yaitu otomatis (sesuai jam yang ditentukan) dan manual (Bluetooth).

II KOMPONEN DAN PENGGUNAANNYA

1. Arduino Nano

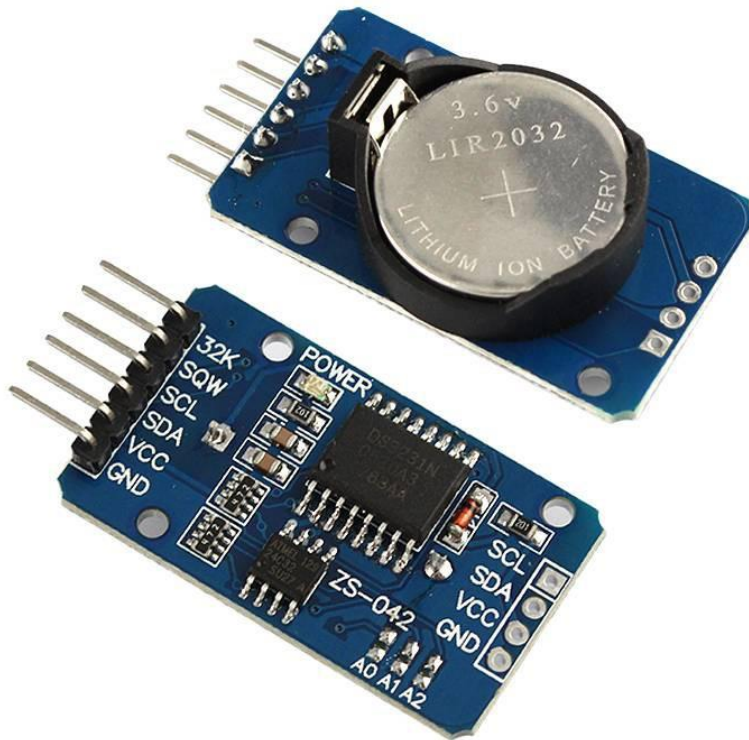


Pemilihan penggunaan Arduino nano sebagai mikrokontroler ini karena Arduino Nano memiliki pin yang banyak, memiliki *build-in voltage regulator*, serta ukurannya yang kecil yang membuat rangkaian utama bisa memiliki ukuran yang lebih kecil juga. Penggunaan Arduino nano adalah sebagai pusat komputasi dan pusat kontrol pada rangkaian yang telah dibuat ini. Arduino nano ini diprogram menggunakan USB Mini-B yang dihubungkan langsung ke komputer. Pemrograman untuk Arduino Nano ini saya lakukan dengan software Arduino IDE. Untuk sumber tegangannya, Arduino nano dapat diberi input Voltase dari 7 – 12 Vdc. Pada rangkaian yang saya buat, sumber tegangan akan berasal dari baterai kotak 9V yang dihubungkan ke modul DC Barrel Jack dan diteruskan ke kaki Vin dan Ground pada Arduino Nano.

2. Modul RTC (*Real Time Clock*) DS3231

RTC (Real Time Clock) merupakan chip IC yang mempunyai fungsi menghitung waktu yang dimulai dari detik, menit, jam, hari, tanggal, bulan, hingga tahun dengan akurat. Untuk menjaga atau menyimpan data waktu yang telah di-ON-kan pada module terdapat sumber catu daya sendiri yaitu baterai jam kancing, serta keakuratan data waktu yang

ditampilkan digunakan osilator kristal eksternal. Sehingga saat perangkat mikrokontroler terhubung dengan RTC ini sebagai sumber data waktu dimatikan, data waktu yang sudah terbaca dan ditampilkan tidak akan hilang begitu saja. Dengan catatan baterai yang terhubung pada RTC tidak habis dayanya.



Pada rangkaian, penggunaan modul RTC DS3231 ini digunakan sebagai sumber data referensi untuk mentrigger mati-nyalanya relay. Pada program yang telah dibuat data yang akan dijadikan referensi adalah jam, menit, dan detik. Sehingga kita juga perlu menginputkan data berupa jam, menit, dan detik, untuk sebagai acuan kapan relay akan ON. Data yang kita inputkan manual nantinya akan terus dibandingkan dalam loop, sehingga saat data yang kita diinputkan sesuai dengan waktu realtime, maka state relay akan ter-trigger.

3. Modul LCD Display 16x2

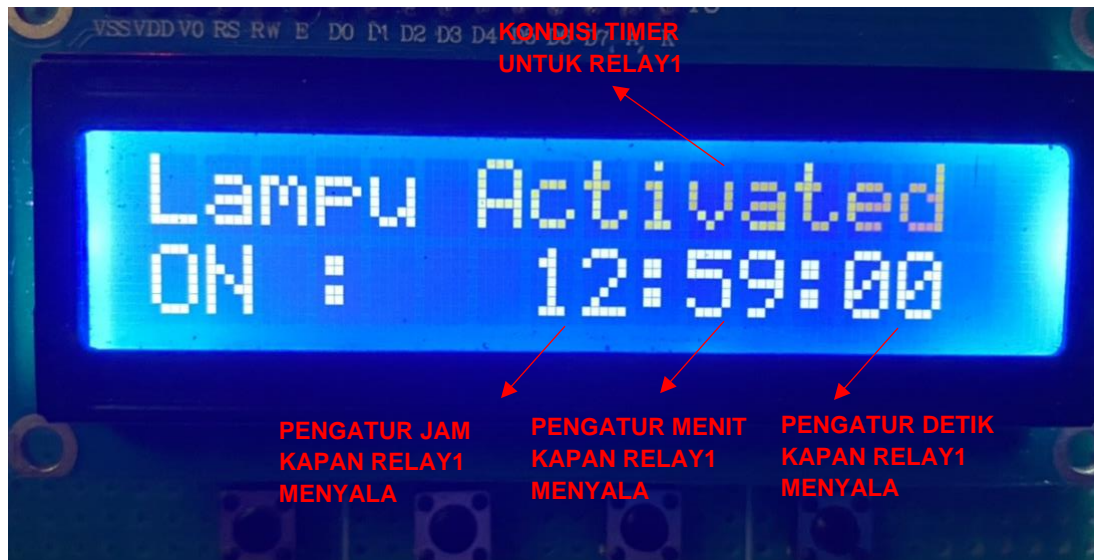
LCD 16×2 (Liquid Crystal Display) merupakan modul penampil data yang mepergunakan kristal cair sebagai bahan untuk penampil data yang berupa tulisan maupun gambar. Pengaplikasian pada kehidupan sehari – hari yang mudah dijumpai antara lain pada kalkulator, gamebot, televisi, atau pun layar komputer. Adapun fitur – fitur yang tersedia antara lain Terdiri dari 16 kolom dan 2 baris, Dilengkapi dengan back light, Mempunyai 192 karakter tersimpan, Dapat dialamati dengan mode 4-bit dan 8-bit, Terdapat karakter

generator terprogram. LCD 16x2 memiliki dua register seperti register data dan register perintah. RS (register select) atau pilihan daftar terutama digunakan untuk mengubah dari satu register ke register lainnya. Ketika set register adalah '0', maka itu dikenal sebagai register perintah. Demikian pula, ketika set register adalah '1', maka itu dikenal sebagai register data. Fungsi utama dari register perintah adalah untuk menyimpan instruksi perintah yang diberikan ke layar. Sehingga tugas yang telah ditentukan dapat dilakukan seperti membersihkan layar, menginisialisasi, mengatur tempat kursor, dan kontrol tampilan. Di sini pemrosesan perintah dapat terjadi dalam register. Fungsi utama daftar data adalah untuk menyimpan informasi yang akan ditampilkan di layar LCD. Di sini, nilai karakter ASCII adalah informasi yang akan ditampilkan di layar LCD. Setiap kali kita mengirim informasi ke LCD, ia mengirimkan ke daftar data, dan kemudian proses akan mulai di sana. Saat daftar diatur = 1, maka daftar data akan dipilih.



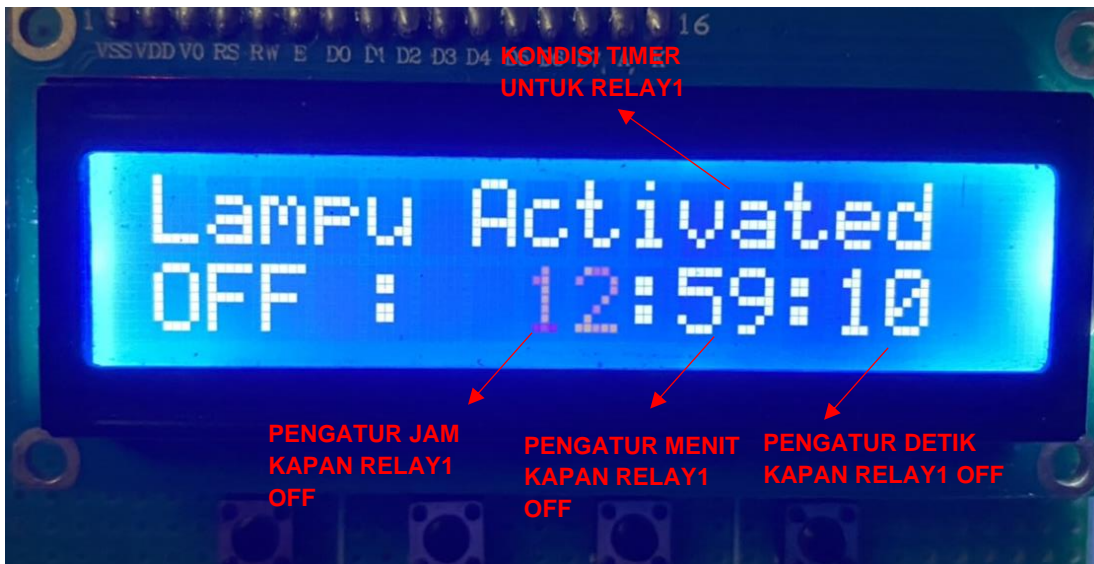
Gambar 3.1 Tampilan Utama LCD

Penggunaan LCD Display 16x2 ini pada rangkaian yaitu sebagai media penampil data yang utamanya dari modul RTC DS3231 yang berupa jam, menit, detik, hari, tanggal, bulan, tahun, serta symbol alarm yang merupakan tanda bahwa timer untuk relay sedang aktif, atau dalam kondisi “activated”. LCD ini juga kurang lebihnya menjadi media *User Interface* (UI) untuk mengatur waktu dari RTC dan juga mengatur timer pada relay seperti pada gambar 3.2 dan gambar 3.3



Gambar 3.2 Tampilan Untuk Setting Waktu ON Relay1

Pada gambar 3.2 merupakan tampilan untuk men-setting waktu kapan relay akan ON. Pada tampilan ini kita dapat mengatur kondisi dari timer itu sendiri, apakah timer itu sedang berjalan (Activated) atau tidak berjalan (Deactivated), dan juga mengatur waktu kapan relay 1 akan menyala.

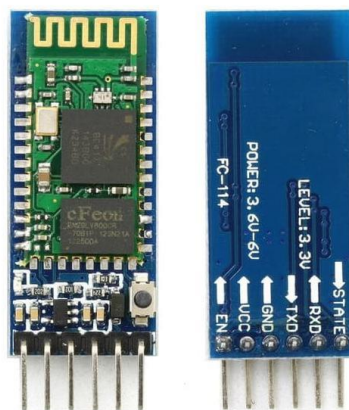


Gambar 3.3 Tampilan Untuk Setting Waktu OFF Relay1

Seperti pada gambar 3.2, tampilan pada gambar 3.5 juga memiliki fungsi yang hampir sama, namun pada tampilan ini kegunaannya adalah untuk men-setting waktu relay1 untuk OFF.

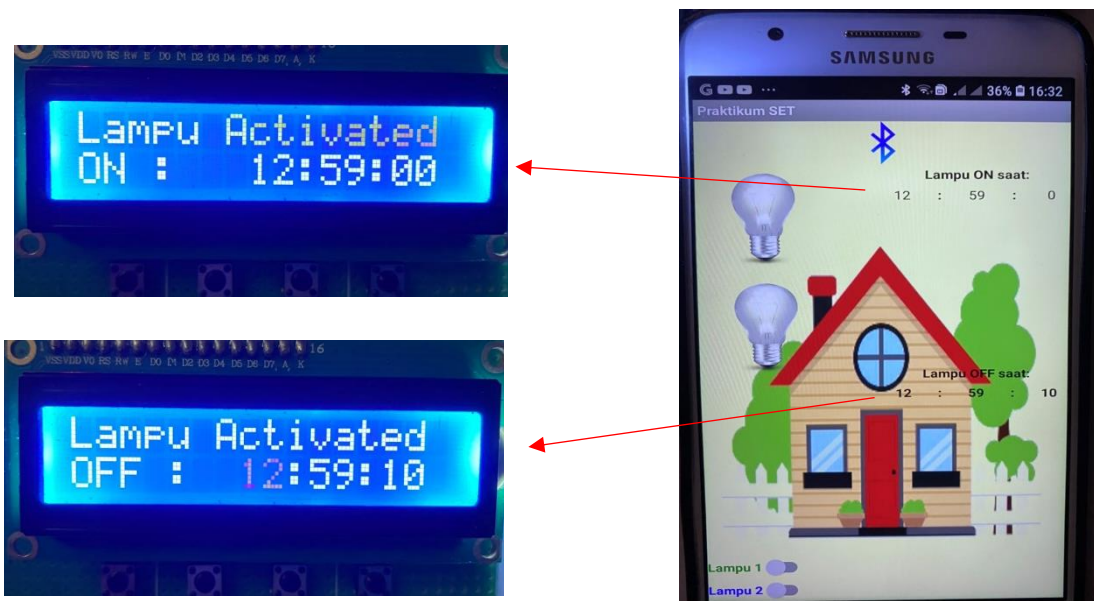
4. Modul Bluetooth HC-05

Module Bluetooth HC-05 adalah modul Bluetooth yang mudah digunakan melalui penggunaan SPP (*Serial Port Protocol*) yang di desain untuk pengaturan koneksi sinyal wireless. Modul ini memenuhi syarat Bluetooth V2.0+EDR (*Enhanced Data Rate*) dengan modulasi sebesar 3 Mbps dan transceiver radio 2,4 GHz. Modul ini menggunakan CSR Bluecore 04-external single chip dengan teknologi CMOS dan Adaptive Frequency Hoping Feature (AFH). Ukuran dari modul ini cukup kecil, yaitu 12,7 mm x 27 mm. Modul Bluetooth HC-05 dengan supply tegangan sebesar 3,3 V ke pin 12 modul Bluetooth sebagai VCC. Pin 1 pada modul Bluetooth sebagai transmitter kemudian pin 2 pada Bluetooth sebagai receiver.



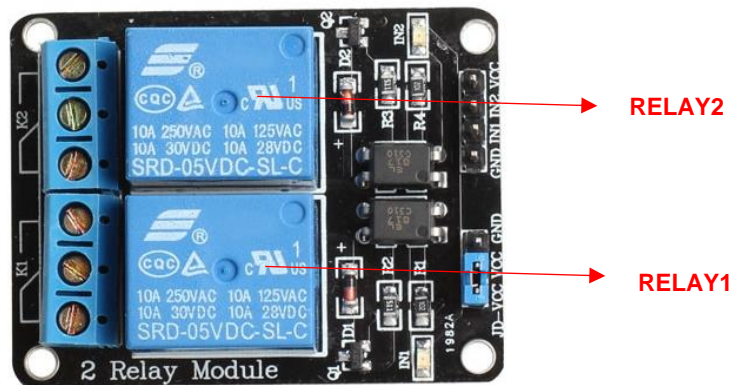
Gambar 4.1 Modul Bluetooth HC-05

Pada rangkaian alat, modul Bluetooth ini digunakan sebagai alat untuk menerima dan mengirim data secara serial ke Handphone yang telah terinstall aplikasi yang telah saya buat. Data yang dikirim adalah memori dari EEPROM yang berisi data berupa jam, menit, dan detik untuk mengatur ON/OFF Relay1. Sedangkan, data yang diterima adalah data berupa flag untuk menyalakan relay1 dan relay2. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



5. Modul Relay 2 Channel

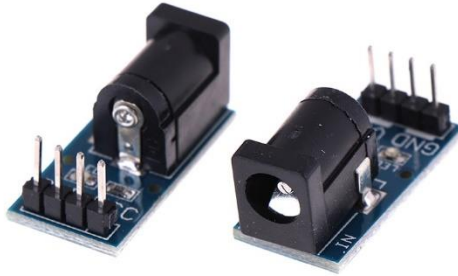
Modul relay adalah salah satu piranti yang beroperasi berdasarkan prinsip elektromagnetik untuk menggerakkan kontaktor guna memindahkan posisi ON ke OFF atau sebaliknya dengan memanfaatkan tenaga listrik. Peristiwa tertutup dan terbukanya kontaktor ini terjadi akibat adanya efek induksi magnet yang timbul dari kumparan induksi listrik. Perbedaan yang paling mendasar antara relay dan sakelar adalah pada saat pemindahan dari posisi ON ke OFF. Relay melakukan pemindahan-nya secara otomatis dengan arus listrik, sedangkan sakelar dilakukan dengan cara manual. Pada dasarnya, fungsi modul relay adalah sebagai saklar elektrik. Dimana ia akan bekerja secara otomatis berdasarkan perintah logika yang diberikan. Kebanyakan, relay 5 volt DC digunakan untuk membuat project yang salah satu komponennya butuh tegangan tinggi atau yang sifatnya AC (Alternating Current). Sedangkan kegunaan relay secara lebih spesifik adalah sebagai berikut: Menjalankan fungsi logika dari mikrokontroler Arduino, Sarana untuk mengendalikan tegangan tinggi hanya dengan menggunakan tegangan rendah, Meminimalkan terjadinya penurunan tegangan, Memungkinkan penggunaan fungsi penundaan waktu atau fungsi time delay function, Melindungi komponen lainnya dari kelebihan tegangan penyebab korsleting, Menyederhanakan rangkaian agar lebih ringkas.



Gambar 5.1 Modul Relay 2 Channel

Pada rangkaian, modul relay ini digunakan sebagai switch untuk menyalakan atau mematikan peralatan elektronik yang ingin kita kontrol. Relay yang berada pada bagian atas (berdekatan dengan pin ke arduino) merupakan Relay2 yang hanya dapat dikontrol melalui data serial atau dengan modul bluetooth. Relay yang disebelahnya merupakan Relay1, yang bisa dikontrol mati nyalanya secara otomatis dengan settingan yang kita buat pada LCD, relay1 juga dapat dikontrol dengan komunikasi serial atau dengan modul bluetooth yang telah tersedia.

6. Modul DC Barrel Jack



Gambar 6.1 Modul DC Barrel Jack

Modul ini merupakan modul tambahan untuk mendapat sumber tegangan dari baterai. Penggunaan modul ini digunakan agar rangkaian bisa menjadi lebih rapih karena kabel dari baterai tidak disolder langsung ke kaki pin Vin Arduino Nano, dan juga memudahkan kita untuk menghubungkan rangkaian dengan beragam jenis baterai atau sumber tegangan yang berbeda.

7. Button Switch SPST



Gambar 7.1 Button Switch SPST

Switch yang digunakan merupakan button switch SPST. Penggunaan button switch ini adalah untuk mengatur waktu dan juga timer untuk menyalakan dan mematikan relay1. Keterangan kegunaan button switch ini dapat dilihat pada gambar 7.2.



Button untuk
mengatur waktu
(SetMode)

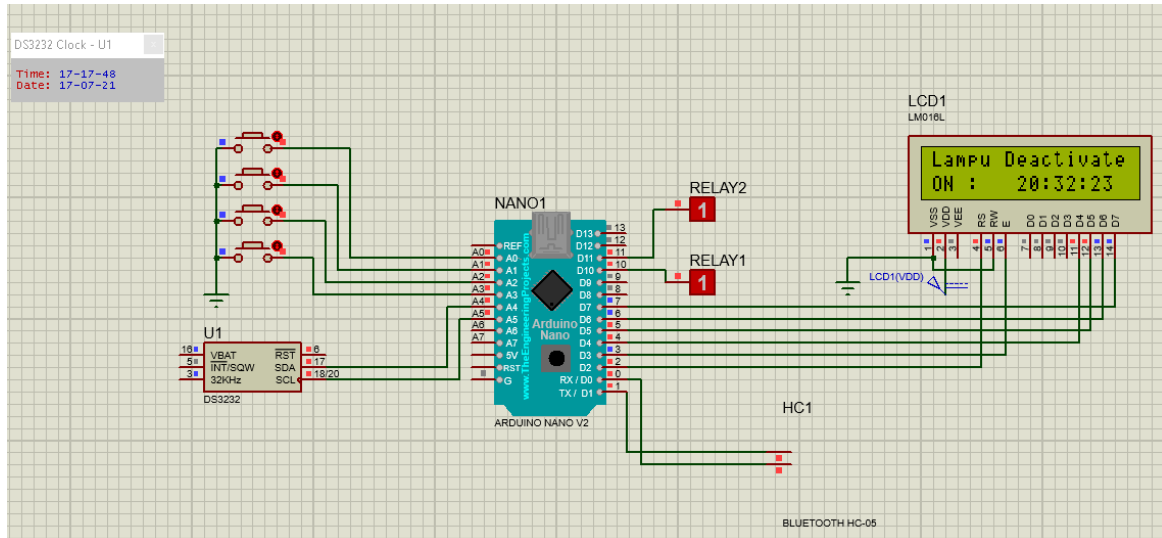
Button untuk
mengatur
increment waktu

Button untuk
mengatur
decrement waktu

Button untuk
mengatur timer

III CARA KERJA ALAT

1. Cara kerja berdasarkan Hardware



Alat ini secara besarnya membutuhkan 1 buah mikrokontroler dan 4 buah modul dan juga 4 buah switch. Hubungan antar komponen jelasnya seperti berikut:

- **RTC DS3231:**
 - VCC -> +5V
 - SCL -> SCL (A5)
 - SDA -> SDA (A4)
 - GND -> GND
- **LCD Display:**
 - VSS -> GND
 - VDD -> +5V
 - V0 -> Trimmer
 - RS -> D2 (Arduino)
 - RW -> GND
 - E -> D3 (Arduino)
 - D4 -> D4 (Arduino)
 - D5 -> D5 (Arduino)
 - D6 -> D6 (Arduino)
 - D7 -> D7 (Arduino)
- **Bluetooth HC-05:**
 - TX -> RX (Arduino)
 - RX -> TX (Arduino)
 - VCC -> +5V
 - GND -> GND

- Button_settime -> A0
- Button_up -> A1
- Button_down -> A2
- Button_timer -> A3

2. Cara kerja berdasarkan Software

Pertama kita akan membutuhkan data dari EEPROM modul RTC (berisi waktu seperti jam, menit, detik, hari, tanggal, bulan, dan tahun) dan juga EEPROM dari Arduino Nano sendiri (berisi mode, AlarmHH, AlarmMM, AlarmSS, AlarmHH2, AlarmMM2, AlarmSS2) yang akan disimpan pada variable yang telah dibuat seperti dibawah ini

```
#include <DS3231.h> //Memanggil RTC3231 Library
#include <Wire.h> // i2C Connection Library
#include <LiquidCrystal.h> //Libraries
#include <EEPROM.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7); //Arduino pins to lcd

#define bt_time A0
#define bt_up A1
#define bt_down A2
#define bt_alarm A3

#define buzzer 8

#define lampul 10
#define lampu2 11

DS3231 rtc(SDA, SCL);

Time t;

int hh = 0, mm = 0, ss = 0, dd = 0, bb = 0, set_day;
int yy = 0;
```

adalah variable untuk menyimpan jam, menit, detik, hari, tanggal, bulan, dan tahun . Digunakan juga fungsi Time t, yang berguna sebagai variable penyimpanan menyimpan jam, menit, detik, hari, tanggal, bulan, dan tahun dari EEPROM modul RTC. Variable Time ini khusus karena 7 data waktu yang didapat dari RTC dapat diparsing sehingga bisa kita masukkan variable waktunya masing-masing (hh,mm,ss,dd,bb,yy) seperti pada gambar di bawah ini.

```
t = rtc.getTime();
Day = rtc.getDOWStr(1);

if (setMode == 0){
  hh = t.hour, DEC;
  mm = t.min, DEC;
  ss = t.sec, DEC;
  dd = t.date, DEC;
  bb = t.mon, DEC;
  yy = t.year, DEC;
}
```

Lalu ada juga variable seperti mode, AlarmHH, AlarmMM, AlarmSS, AlarmHH2, AlarmMM2, AlarmSS2, setMode, setAlarm, dan alarmMode, yang fungsinya adalah seperti berikut:

setMode = keterangan flag set waktu, 0(tidak sedang setwaktu), 1(set jam), 2(set menit), 3(set detik), 4(set hari), 5(set tanggal), 6(set bulan), 7(set tahun), 8(set time finished)

setAlarm = keterangan flag set alarm, 0(tidak sedang setalarm), 1(set activate/deactivate), 2(set jam ON), 3(set menit ON), 4(set detik ON), 5(set jam OFF), 6(set menit OFF), 7(set detik OFF), 8(set alarm finished)

alarmMode = keterangan alarm 1(alarm sudah pernah disetting walaupun itu activate/deactivate), 0(alarm belum pernah tersentuh/tersetting)

mode = keterangan dari memori apakah alarm activate(1) / deactivate(0)

flag = status button time dan button alarm, press(0) / idle(1)

```
int AlarmHH = 21, AlarmMM = 22, AlarmSS = 23, setMode = 0, setAlarm = 0, alarmMode=0;
int AlarmHH2 = 22, AlarmMM2 = 23, AlarmSS2 = 24;
int stop =0, mode=0, flag=0;
```

Dari sini, kita sudah mendapatkan data-data yang dibutuhkan, maka selanjutnya kita tinggal print data-data tersebut pada LCD.

```
if(setAlarm==0){
    lcd.setCursor(4,0);
    lcd.print((hh/10)%10);
    lcd.print(hh % 10);
    lcd.print(":");
    lcd.print((mm/10)%10);
    lcd.print(mm % 10);
    lcd.print(":");
    lcd.print((ss/10)%10);
    lcd.print(ss % 10);
    lcd.print(" ");
    if(mode==1){lcd.write(2);}
    else{lcd.print(" ");}
    lcd.print(" ");

    lcd.setCursor(1,1);
    lcd.print(Day);
    lcd.print(" ");
    lcd.print((dd/10)%10);
    lcd.print(dd % 10);
    lcd.print("/");
    lcd.print((bb/10)%10);
    lcd.print(bb % 10);
    lcd.print("/");
    lcd.print((yy/1000)%10);
    lcd.print((yy/100)%10);
    lcd.print((yy/10)%10);
    lcd.print(yy % 10);
}
```


Disini semua program untuk menampilkan data ke LCD berada dalam Loop If(setAlarm ==), Ini memastikan kita tidak dalam mode setting timer untuk relay. Seperti yang kita tahu setAlarm adalah flag saat mensetting timer pada relay yang terdiri dari 1 sampai 8 kode flag. Kita memastikan kode flag setAlarm == 0, karena hanya pada mode setAlarm tampilan LCD signifikan berbeda daripada tampilan utama dan juga tampilan saat setMode (set waktu).

```

setupClock();
setTimer();
delay (100);
blinking();
//Serial.print("timer |");
Serial.print(AlarmHH); Serial.print("|"); Serial.print(AlarmMM); Serial.print("|"); Serial.print(AlarmSS); Serial.print("|");
Serial.print(AlarmHH2); Serial.print("|"); Serial.print(AlarmMM2); Serial.print("|"); Serial.print(AlarmSS2); Serial.print("|");
relay1();
relay2();
delay (100);

```

Masih dalam void loop(), selanjutnya kita akan masuk ke fungsi setupClock() yang berisi beberapa definisi dari tiap flag 1 sampai 8 pada setMode (set waktu) dan setAlarm (set timer relay1), dan juga fungsi-fungsi dari setiap button. Selanjutnya kita juga masuk ke fungsi setTimer() yang berisi mengganti tampilan LCD ke tampilan untuk setting timer relay1.

```

void setTimer () {
    //Alarm
    if (setMode == 0 && setAlarm >0 && setAlarm <5) {
        lcd.setCursor (0,0);
        lcd.print("Lampu ");
        if(mode==0){lcd.print("Deactivate");}
        else{lcd.print("Activated ");}
        lcd.setCursor (0,1);
        lcd.print("ON : ");
        lcd.setCursor (7,1);
        lcd.print((AlarmHH/10)%10);
        lcd.print(AlarmHH % 10);
        lcd.print(":");
        lcd.print((AlarmMM/10)%10);
        lcd.print(AlarmMM % 10);
        lcd.print(":");
        lcd.print((AlarmSS/10)%10);
        lcd.print(AlarmSS % 10);
    }
    //*****//*****//*****//*****//*****//*****
    if (setMode == 0 && setAlarm >4 && setAlarm <8) {
        lcd.setCursor (0,0);
        lcd.print("Lampu ");
        if(mode==0){lcd.print("Deactivate");}
        else{lcd.print("Activated ");}
        lcd.setCursor (0,1);
        lcd.print("OFF : ");
        lcd.setCursor (7,1);
        lcd.print((AlarmHH2/10)%10);
        lcd.print(AlarmHH2 % 10);
        lcd.print(":");
        lcd.print((AlarmMM2/10)%10);
        lcd.print(AlarmMM2 % 10);
        lcd.print(":");
        lcd.print((AlarmSS2/10)%10);
        lcd.print(AlarmSS2 % 10);
    }
    //*****//*****//*****//*****//*****//*****
}
//-----

```

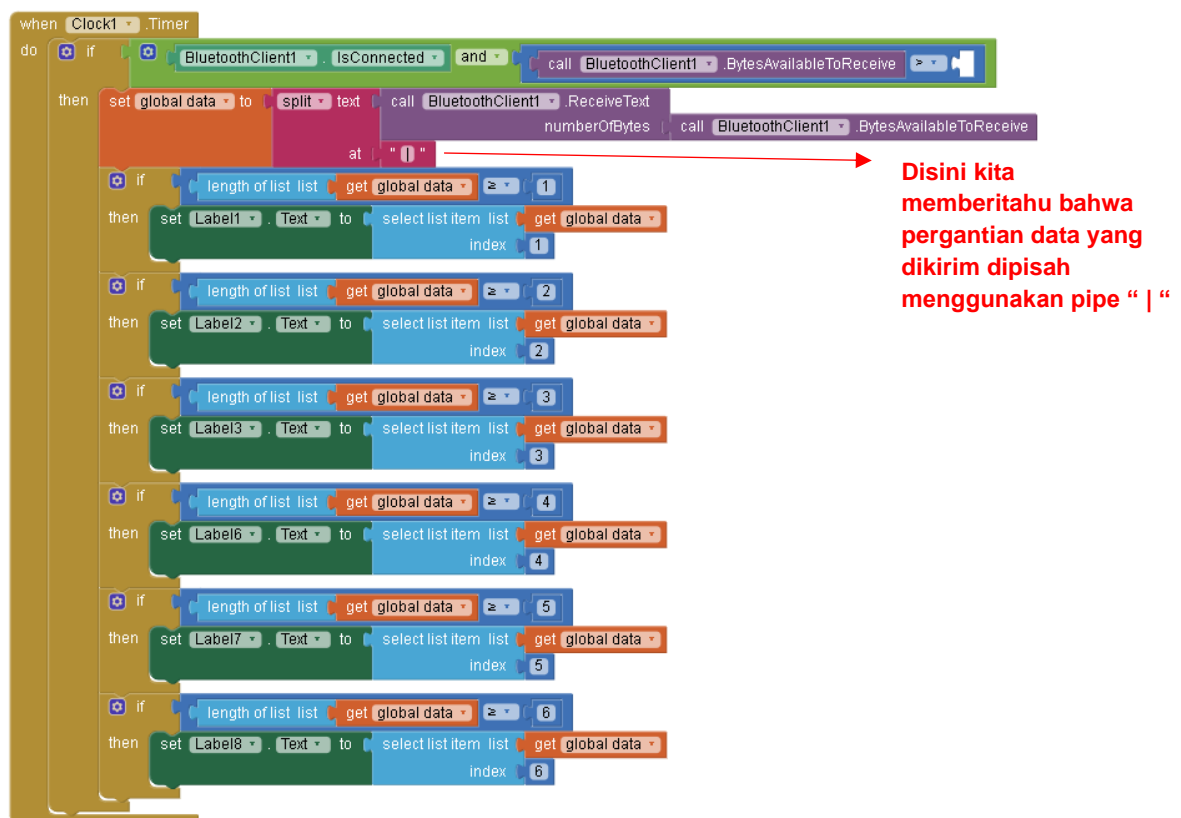
Selanjutnya adalah pemanggilan loop blinking() yang merupakan loop untuk membuat efek blinking pada setMode ataupun setAlarm, sehingga kita bisa mengetahui komponen apa yang sedang ingin kita ubah. Void blinking mudahnya adalah penghapusan tulisan per komponen dan setelah itu terisi kembali tulisan komponen tersebut karena program diatasnya, yang

dilakukan secara berulang-ulang karena masuk dalam void loop(). Program void blinking kurang lebihnya seperti pada gambar dibawah ini.

```
void blinking (){
//BLINKING SCREEN
if (setAlarm <2 && setMode == 1){lcd.setCursor(4,0); lcd.print(" ");}
if (setAlarm <2 && setMode == 2){lcd.setCursor(7,0); lcd.print(" ");}
if (setAlarm <2 && setMode == 3){lcd.setCursor(10,0); lcd.print(" ");}
if (setAlarm <2 && setMode == 4){lcd.setCursor(1,1); lcd.print(" ");}
if (setAlarm <2 && setMode == 5){lcd.setCursor(5,1); lcd.print(" ");}
if (setAlarm <2 && setMode == 6){lcd.setCursor(8,1); lcd.print(" ");}
if (setAlarm <2 && setMode == 7){lcd.setCursor(11,1); lcd.print(" "); }
//Alarm
if (setMode == 0 && setAlarm == 1){lcd.setCursor(6,0); lcd.print(" "); }
if (setMode == 0 && setAlarm == 2){lcd.setCursor(7,1); lcd.print(" "); }
if (setMode == 0 && setAlarm == 3){lcd.setCursor(10,1); lcd.print(" "); }
if (setMode == 0 && setAlarm == 4){lcd.setCursor(13,1);lcd.print(" "); }

if (setMode == 0 && setAlarm == 5){lcd.setCursor(7,1); lcd.print(" "); }
if (setMode == 0 && setAlarm == 6){lcd.setCursor(10,1); lcd.print(" "); }
if (setMode == 0 && setAlarm == 7){lcd.setCursor(13,1);lcd.print(" "); }
}
```

Selanjutnya adalah kita mengirim 6 buah data dengan komunikasi serial, data ini berisi memori terakhir data dari AlarmHH, AlarmMM, AlarmSS, AlarmHH2, AlarmMM2, AlarmSS2. Karena pada system komunikasi serial data yang dikirim dengan sistem *burst* (tidak ada penanda). Disini saya menggunakan metode parsing data menggunakan pipe “|”, sehingga program pada aplikasi android dapat secara otomatis mengenali data yang dikirim dengan metode ini. Program pada software android itu sendiri dapat dilihat pada gambar dibawah ini.



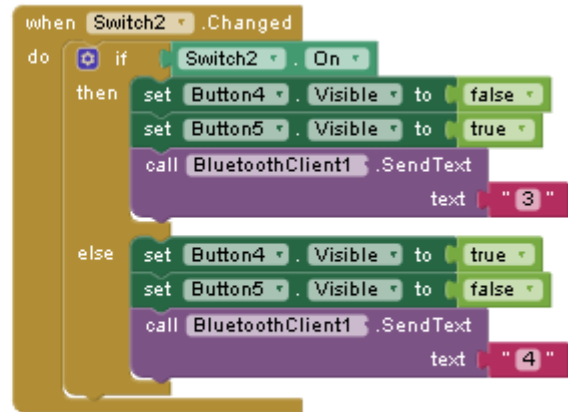
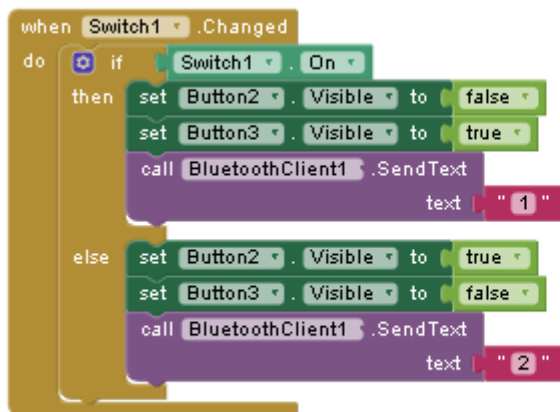
Selanjutnya pada void loop kita akan memasuki 2 loop yang hamper mirip, yaitu fungsi relay1() dan relay2(). Kedua fungsi ini sama-sama memiliki kegunaan yang sama yaitu untuk mengatur waktu, loop relay1() mempunyai beberapa logika untuk mengkontrol relay1, dan loop relay2() mempunyai beberapa logika untuk mengkontrol relay2. Pada loop relay1(), digunakan beberapa logika if dan if else, karena relay1 dapat dikontrol secara otomatis sesuai waktu yang kita atur pada setAlarm dan juga dari komunikasi serial (Bluetooth). Hampir sama dengan loop relay1(), loop relay2() juga memiliki beberapa logika if dan if else, hanya saja relay2 diatur hanya dapat dikontrol dengan komunikasi serial (sehingga tidak dapat ter-trigger secara otomatis).

```
void relay1(){
    if (alarmMode==1 && mode==1 && hh==AlarmHH && mm==AlarmMM && ss==AlarmSS) {
        lcd.clear();
        lcd.setCursor (0,0);
        lcd.print("Lampu Akan On : ");
        lcd.setCursor (7,1);
        lcd.print("3");
        delay(1000);
        lcd.setCursor (7,1);
        lcd.print("2");
        delay(1000);
        lcd.setCursor (7,1);
        lcd.print("1");
        delay(1000);
        digitalWrite(lampul, LOW);
        lcd.clear();
    }
    else if (alarmMode==1 && mode==1 && hh==AlarmHH2 && mm==AlarmMM2 && ss==AlarmSS2) {
        lcd.clear();
        lcd.setCursor (0,0);
        lcd.print("Lampu Akan OFF : ");
        lcd.setCursor (7,1);
        lcd.print("3");
        delay(1000);
        lcd.setCursor (7,1);
        lcd.print("2");
        delay(1000);
        lcd.setCursor (7,1);
        lcd.print("1");
        delay(1000);
        digitalWrite(lampul, HIGH);
        lcd.clear();
    }
    else if (data == '3') {digitalWrite(lampul, LOW);}
    else if (data == '4') {digitalWrite(lampul, HIGH);}
}

void relay2() {
    if (data == '1') {digitalWrite(lampu2, LOW);}
    else if (data == '2') {digitalWrite(lampu2, HIGH);}
}
```

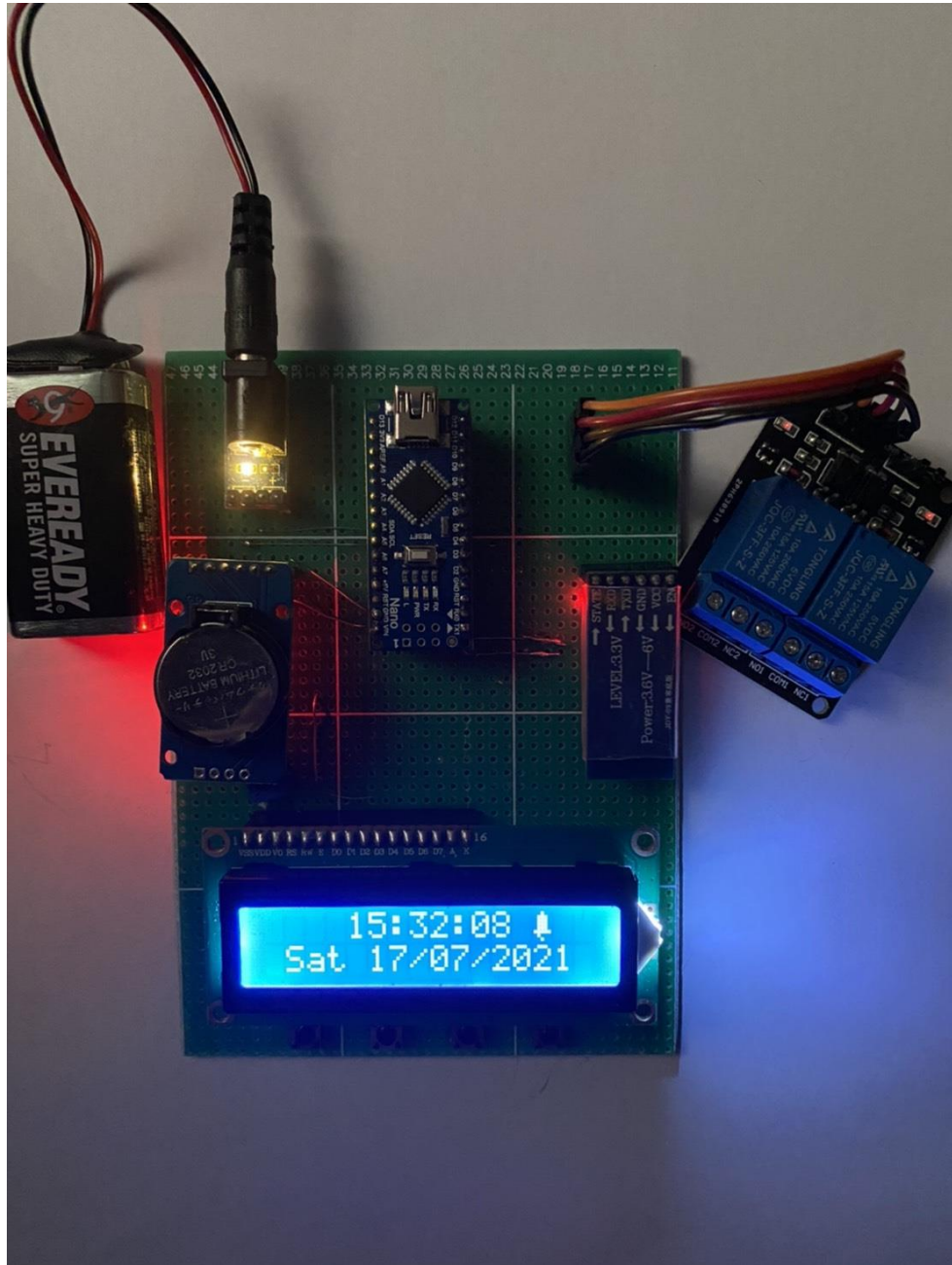
Bisa dilihat, disini ada beberapa state yang bertipe data char dari 1,2,3,4. Data ini didapatkan dari komunikasi serial yang dikirim dari program yang ada pada aplikasi di android. Program

untuk mengirim tipe data char pada aplikasi tersebut kurang lebihnya seperti gambar di bawah ini.



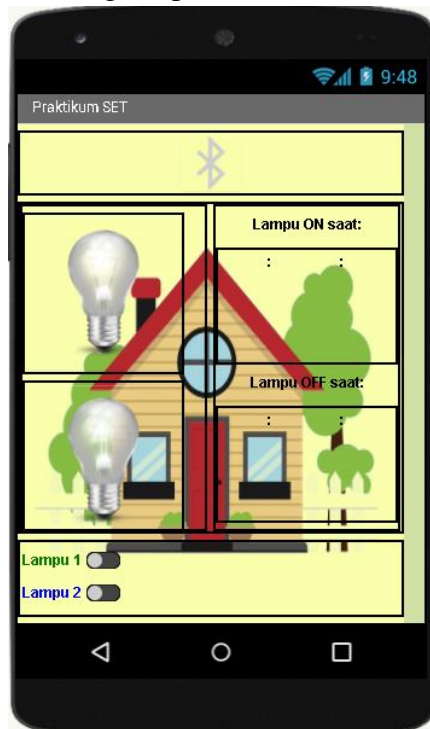
IV FOTO ALAT

Rangkaian alat yang telah dibuat:

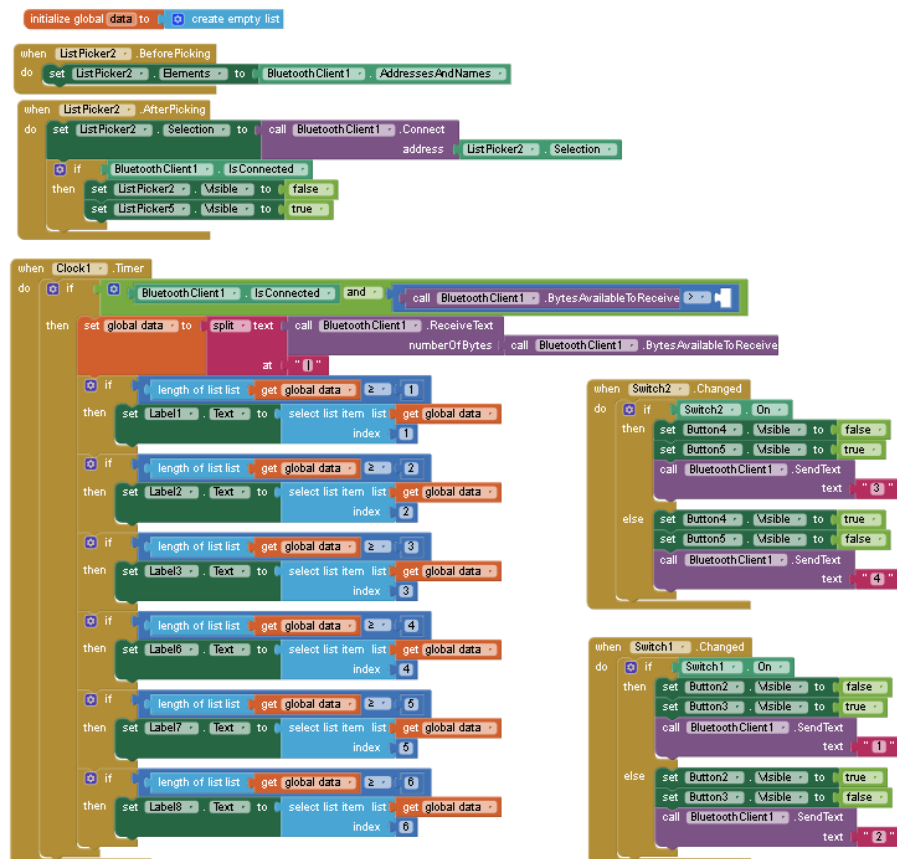


Aplikasi pada android yang telah dibuat:

- Rancangan aplikasi:



- Program untuk aplikasi:



- Tampilan Aplikasi pada android:

