

PANDUAN PEMBUATAN TELEDIAGNOSTIC NONINVASIVE BLOOD GLUCOSE (TENABLE)

Oleh:

Siti Badriah

Yayan Bahtiar

Asep Andang



Daftar Isi

Daftar Isi	i
A. Pendahuluan	1
1. Cara kerja	1
2. Goals projek.....	1
B. Pembahasan	2
1. Alat dan bahan yang diperlukan.....	2
a. Alat yang digunakan sebagai penunjang dalam pembuatan device TENABLE	2
b. Bahan yang diperlukan untuk satu set device TENABLE	2
2. Desain PCB	4
a. Rangkaian skematik jam tangan TENABLE	4
b. Desain PCB jam tangan TENABLE	8
c. Rangkaian skematik cincin TENABLE.....	10
d. Desain PCB cincin TENABLE	11
e. Hasil print PCB	13
f. Hasil pemasangan komponen pada PCB.....	14
3. Desain case	15
a. Desain case jam tangan TENABLE.....	15
b. Desain case cincin TENABLE	17
c. Hasil print 3D case jam tangan dan cincin.....	19
4. Hasil Akhir.....	20
5. Program device TENABLE.....	22
a. Isi Program	22
b. Penjelasan Program.....	26
6. Testing	35
C. Kesimpulan.....	38
D. Dokumentasi Kerja	38

A. Pendahuluan

Telediagnostic Non-Invasive Blood Glucose atau disingkat menjadi TENABLE merupakan sebuah device pengukur kandungan glukosa. TENABLE memiliki keunggulan karena dibuat dalam bentuk jam tangan sehingga nyaman digunakan dan dapat memonitoring kandungan gula darah secara real time. Pada versi 1 ini jam tangan TENABLE memiliki sensor yang disematkan pada sebuah cincin yang tersambung langsung dengan kabel.

1. Cara kerja

Sensor yang disematkan pada cincin yang tersambung dengan jam tangan TENABLE adalah sensor Near Infrared (NIR) 1050nm dan fotodioda yang dipasang secara sejajar. Near Infrared memancarkan sinar infrared dalam jarak dekat menembus kulit dan melewati pembuluh darah kemudian dipantulkan kembali oleh tulang. Pancaran sinar infrared yang telah dipantulkan kemudian diterima oleh fotodioda dan diubah menjadi sinyal listrik.

Intensitas sinar infrared yang telah diterima oleh fotodioda dapat dipengaruhi oleh molekul glukosa pada pembuluh darah yang dapat melakukan penyerapan atau pemantulan terhadap sinar tersebut. Sehingga hal ini menjadi dasar dalam pengukuran kandungan glukosa pada jam tangan TENABLE.

2. Goals projek

Projek TENABLE versi 1 ini memiliki goals untuk dicapai diantaranya,

- a. Pembuatan PCB
- b. Pembuatan case jam tangan dan cincin TENABLE
- c. Perakitan sistem
- d. Upload program
- e. Sistem bekerja dengan membaca input sensor dalam bentuk ADC

B. Pembahasan

1. Alat dan bahan yang diperlukan

a. Alat yang digunakan sebagai penunjang dalam pembuatan device

TENABLE

No	Nama Barang
1	Solder
2	Solder Uap (Blower)
3	Tweezer
4	Tang Potong
5	Multimeter
6	Amplas
7	Pembersih Solder
8	Penyedot Timah
9	Solder Wick
10	Flux
11	Cairan Tiner
12	Kuas

Tabel 1. Alat yang digunakan

b. Bahan yang diperlukan untuk satu set device TENABLE

No.	Komponen	Nilai	Jumlah
Jam Tangan TENABLE			
1	Resistor	1K	4
		1.2K	1
		100	1

		1M	2
		50K	2
		680K	1
		68K	1
		220	1
		50K	1
2	Kapasitor	100n	3
		10u	1
3	LED		2
4	Port USB Micro Female		1
5	TP4056		1
6	FS8205A		1
7	DW01A		1
8	LM358		1
9	Port Jack Audio		1
10	ESP32 WROOM32		1
11	LCD TFT 1.8 Inch		1
12	Baterai Li ion 3.7V Rechargeable		2
13	Set Case Jam Tangan		1
14	Strap Jam Tangan		1
Cincin TENABLE			
1	NIR LED 1050nm		1
2	Port jack Audio female		1
3	Fotodioda		1

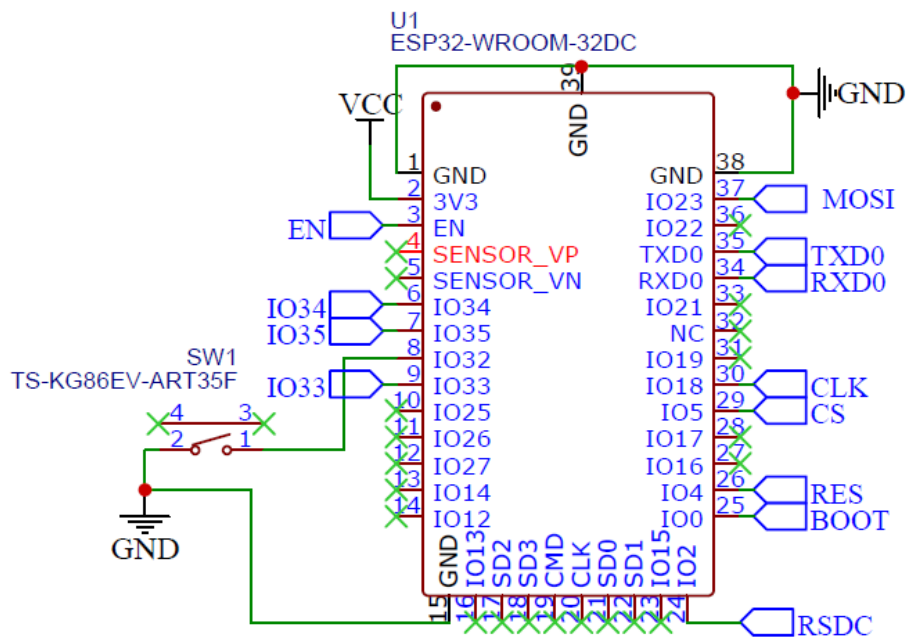
4	Kabel AUX Jack 3.5 mm		1
5	Set Case Cincin		1
6	Timah Cair		1

Tabel 2. Bahan yang diperlukan untuk satu device TENABLE

2. Desain PCB

a. Rangkaian skematik jam tangan TENABLE

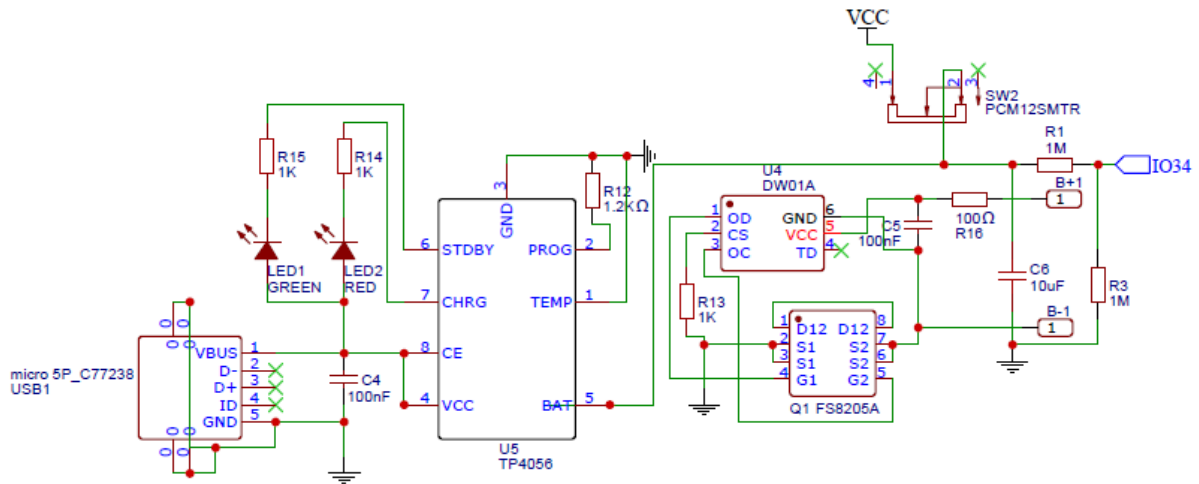
i. Rangkaian Utama



Gambar 1. Skematik Rangkaian Utama

Rangkaian Utama terdiri dari ESP32 yang menjadi kontrol utama device TENABLE.

ii. Rangkaian Charging



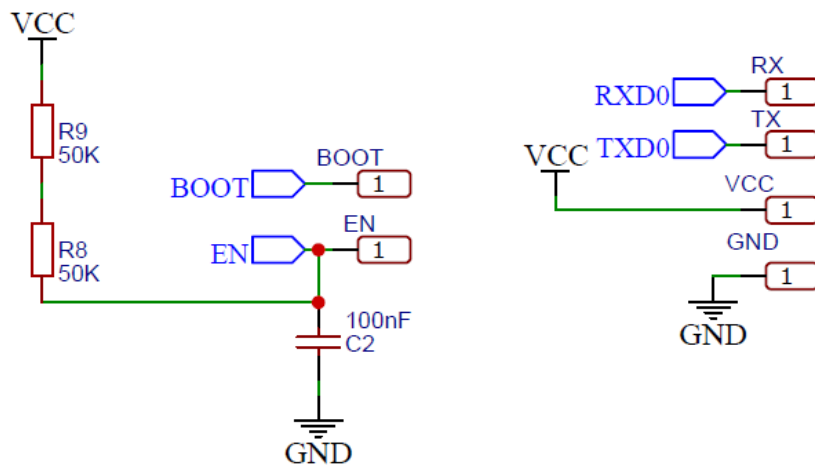
Gambar 2. Skematik Rangkaian Charging

Rangkaian Charging dibuat dengan tujuan untuk mengatur proses pengecasan ke baterai dan mengatasi over charge pada baterai. Selain itu adapun sambungan ke IO34 sebagai indikator kapasitas baterai.

Catatan :

- Ada kesalahan pada rangkaian charging yaitu kutub positif baterai tidak tersambung ke VCC sehingga baterai tidak dapat dicharge maupun mensupply rangkaian.
- Ada masalah dalam mendeteksi kapasitas baterai sehingga perubahan kapasitas baterai tidak terdeteksi perubahannya oleh IO34 ESP32.

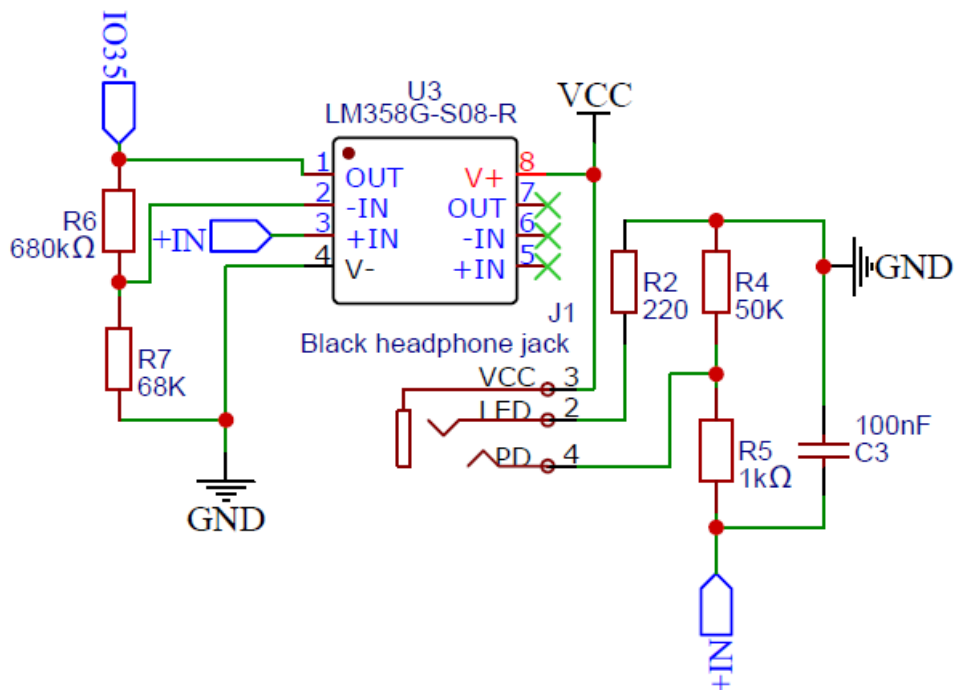
iii. Rangkaian Uploader



Gambar 3. Skematik Rangkaian Uploader

Rangkaian Uploader digunakan untuk mengupload program pada pertama kali device TENABLE dibuat.

iv. Rangkaian Penunjang Sensor



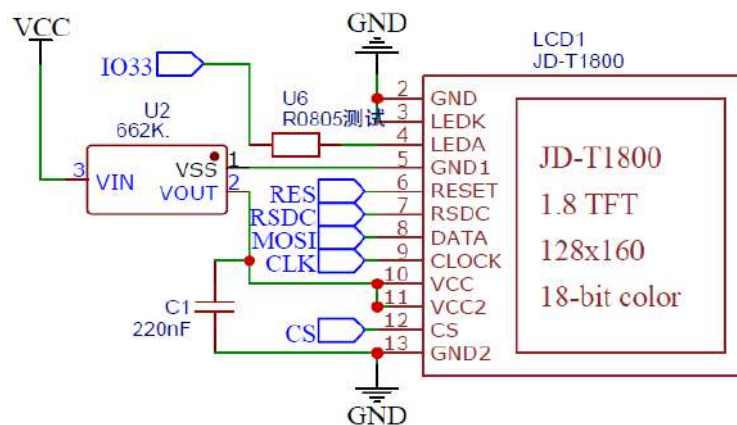
Gambar 4. Skematik Rangkaian Penunjang Sensor

Rangkaian penunjang sensor meliputi rangkaian input, output dan OP AMP. Rangkaian input menggunakan rangkaian resistor PULL DOWN.

Catatan :

- Penggunaan nilai resistor pada R4 harus ditinjau kembali. Developer TENABLE V1 menyarankan kedepannya agar nilai resistor dimulai dari 100K sampai 1M Ohm. Karena penggunaan Resistor 50K pada R4 mempengaruhi sinyal input yang terlalu kecil meskipun sudah menggunakan OP AMP sebagai penguatan sinyal.
- IC OP AMP seharusnya diberikan supply tegangan (V+) sebesar 5V, karena penguatan maksimal pada OP AMP besarnya $V+ - 1.7 V$. Artinya ketika V+ diberikan tegangan 3.7 V maksimal penguatannya adalah 2V. Hal ini mengakibatkan sinyal yang masuk ke ESP32 nilai input maksimalnya dibatasi pada 2V.

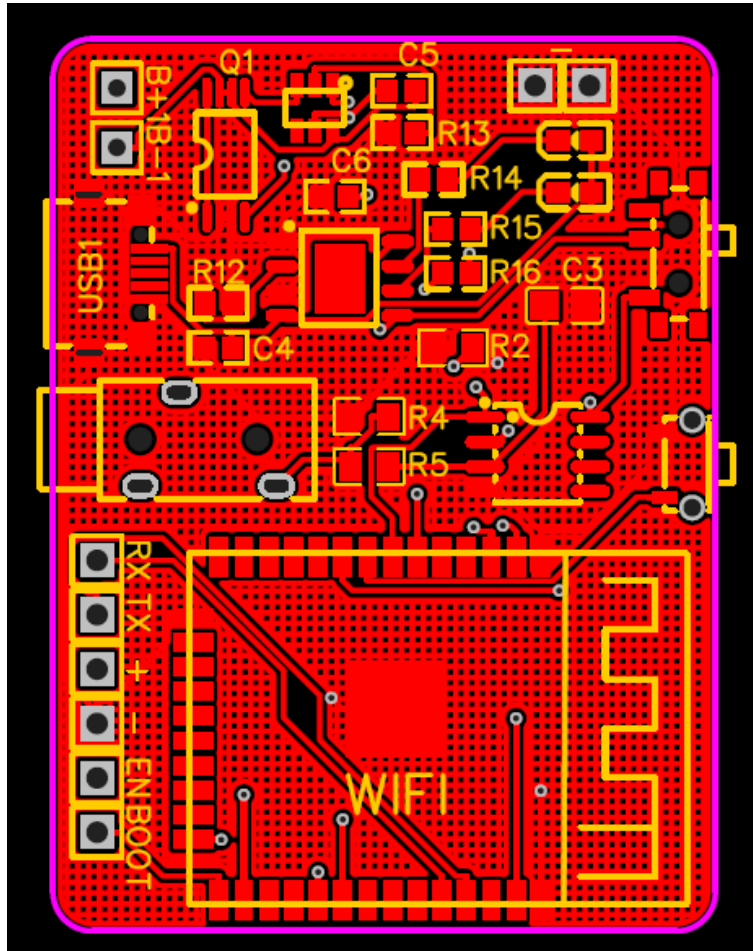
v. Rangkaian Penunjang LCD



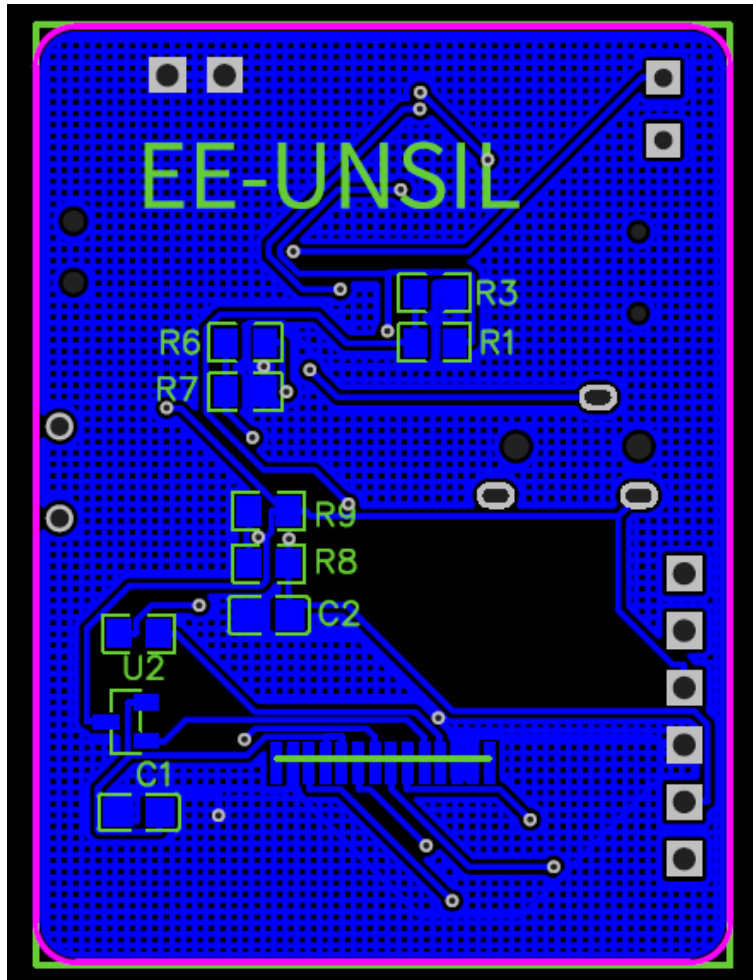
Gambar 5. Skematik Rangkaian Penunjang LCD

Rangkaian ini adalah rangkaian tambahan agar LCD TFT dapat bekerja sebagai mestinya.

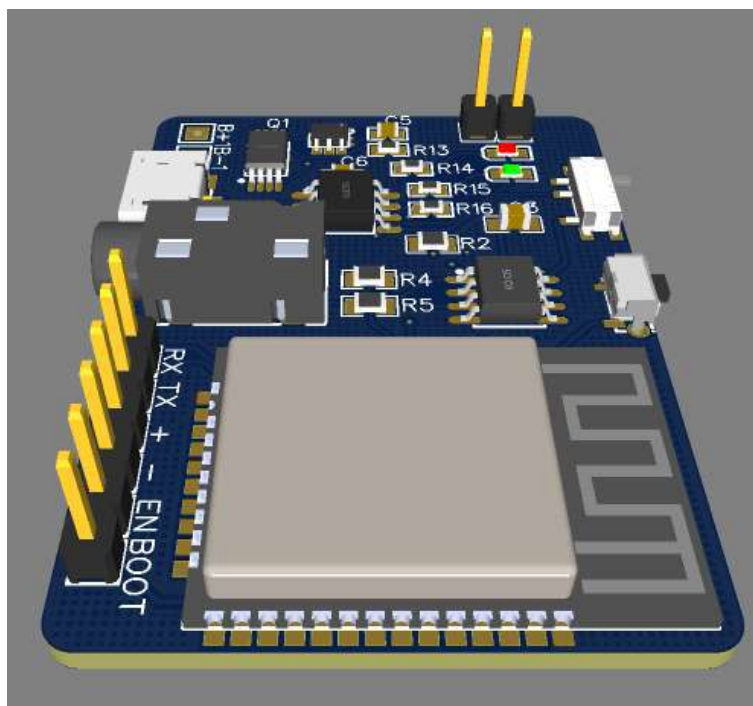
b. Desain PCB jam tangan TENABLE



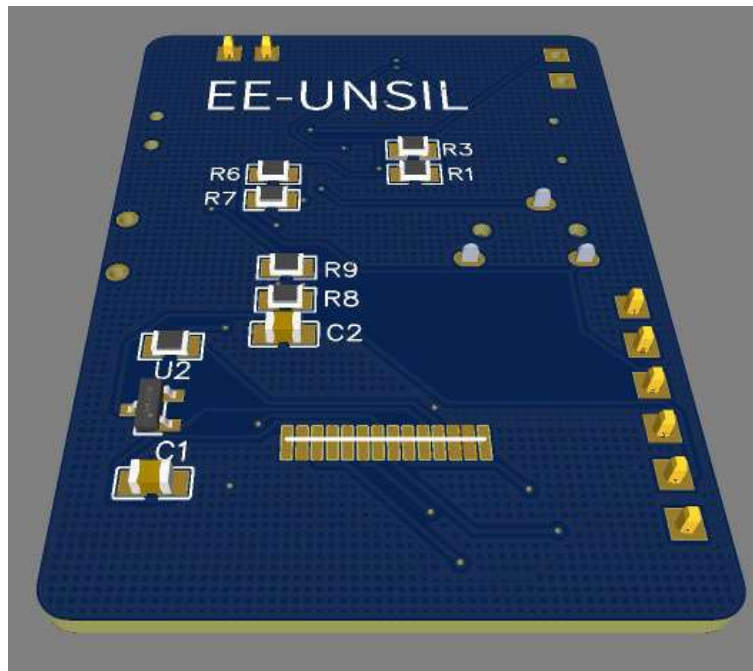
Gambar 6. Desain PCB Top Layer



Gambar 7. Desain PCB Bottom Layer



Gambar 8. Model 3D Desain PCB Top Layer



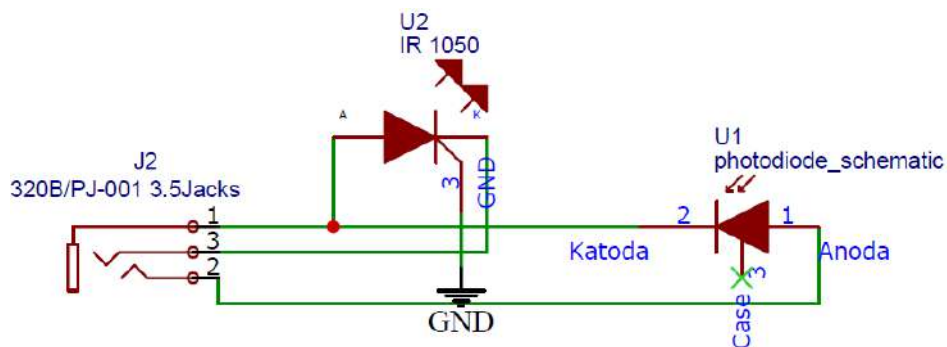
Gambar 9. Model 3D Desain PCB Bottom Layer

Proses pembuatan model PCB dilakukan terlebih dahulu sebelum desain case jam tangan. Sehingga bentuk jam tangan mengikuti bentuk PCB.

Catatan :

Supaya bentuk case jam tangan memiliki bentuk desain yang lebih menarik maka proses desain jam tangan dilakukan terlebih dahulu dan model PCB mengikuti bentuk jam tangan.

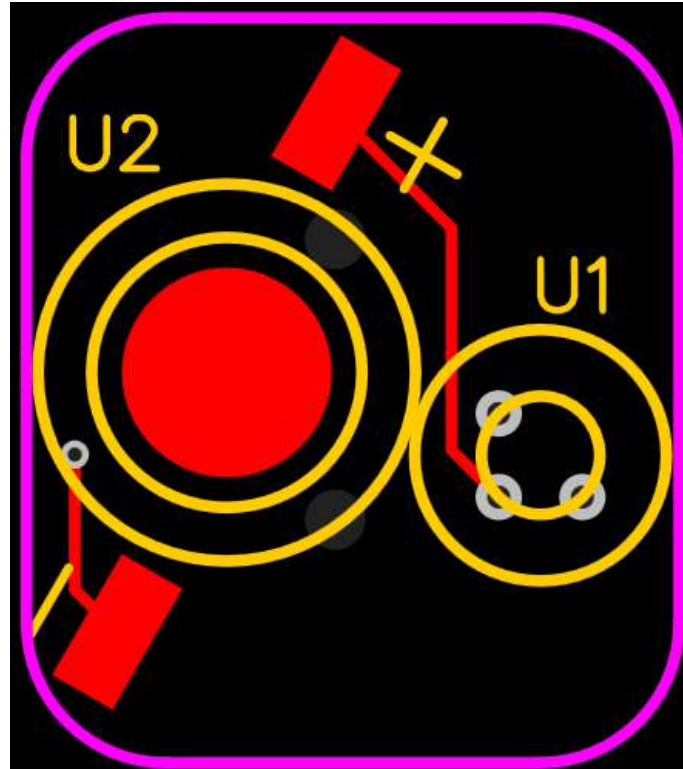
c. Rangkaian skematik cincin TENABLE



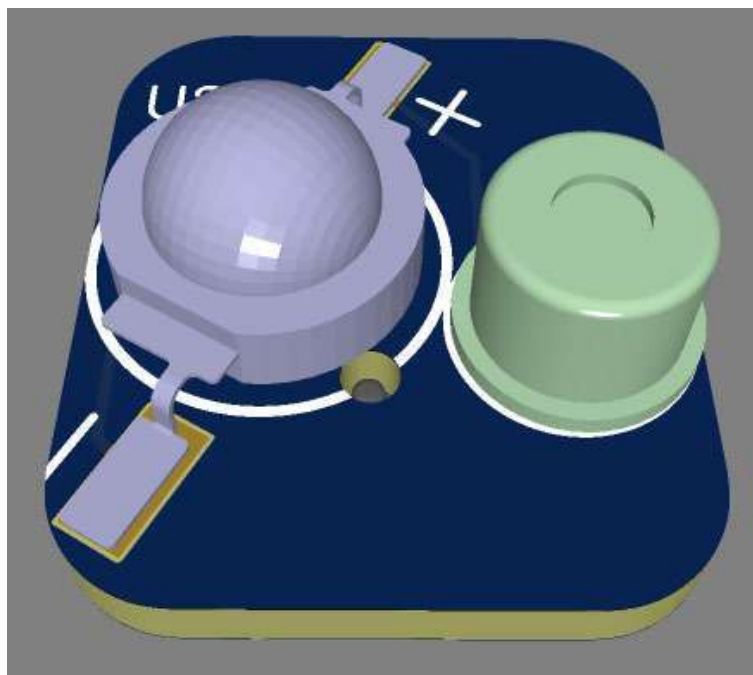
Gambar 10. Skematik Cincin TENABLE

Rangkaian cincin TENABLE terdiri dari 3 komponen yaitu female jack audio 3.5 mm.

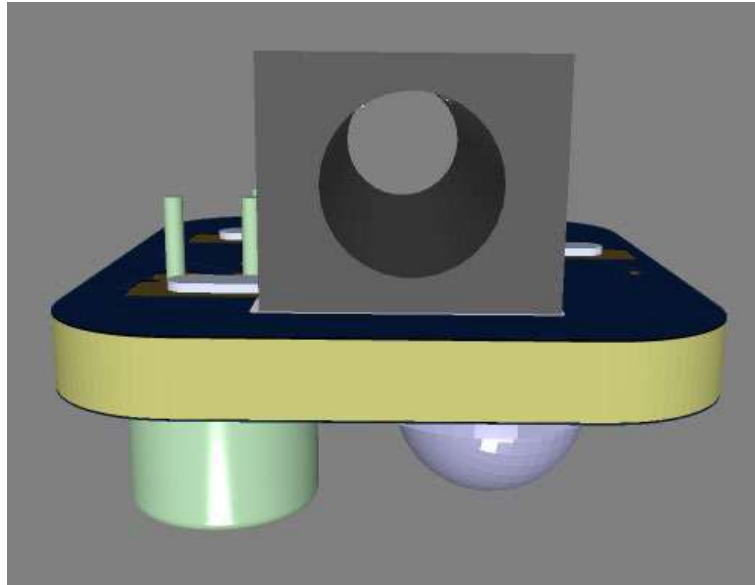
d. Desain PCB cincin TENABLE



Gambar 11. Desain PCB Cincin



Gambar 12. Model 3D Desain PCB Top



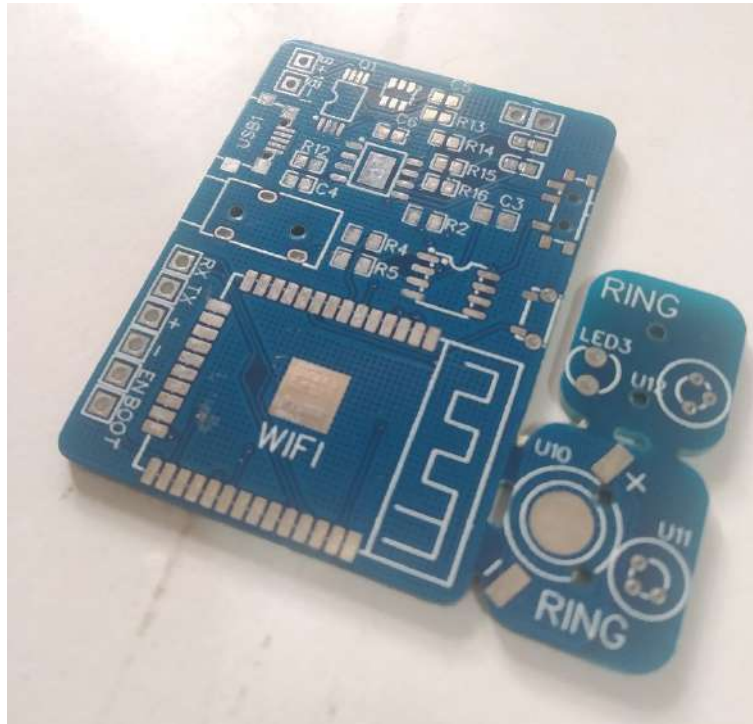
Gambar 13. Model 3D Desain PCB Bottom

Model PCB cincin TENABLE hanya dapat digunakan dengan Infrared 1050 nm.

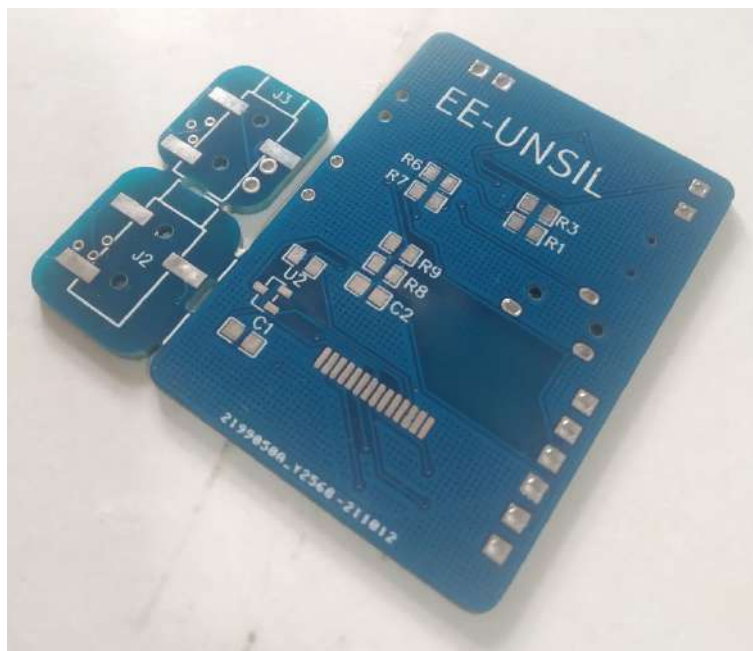
Catatan :

Perlu ditinjau kembali penggunaan Infrared dengan spesifikasi panjang gelombang yang berbeda misalnya 1550 nm. Hal ini berdampak pada berbedanya model PCB karena kaki IR 1050 nm berbeda dengan kaki IR dengan spesifikasi panjang gelombang yang berbeda.

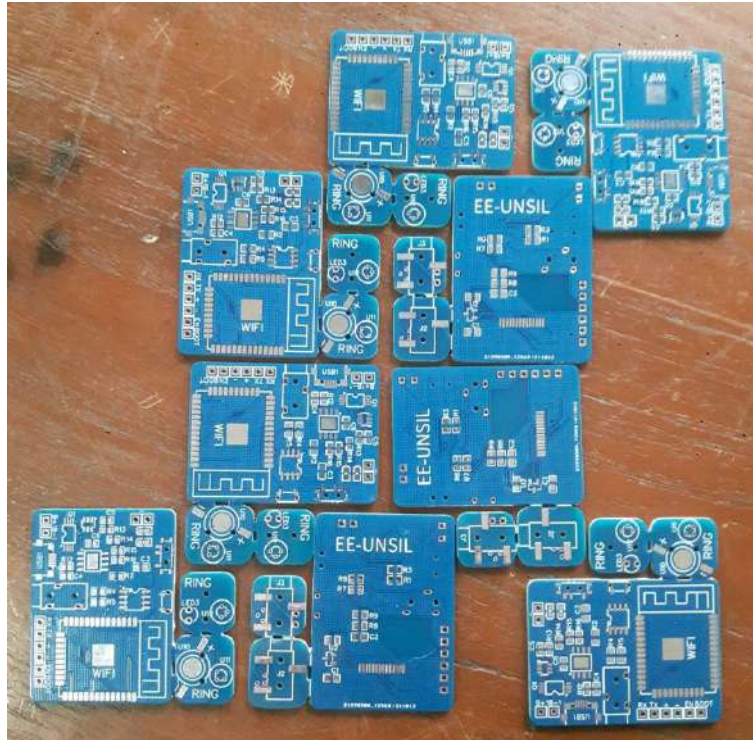
e. Hasil print PCB



Gambar 14. Hasil Cetak PCB Top



Gambar 15. Hasil Cetak PCB Bottom



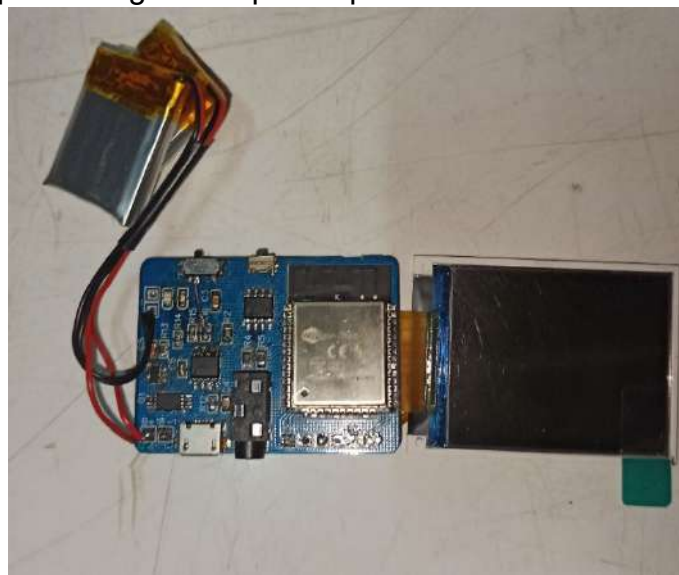
Gambar 16. Sampel Beberapa Hasil Cetak PCB

Hasil print PCB jam tangan dan cincin disatukan agar efisien dalam pemesanan PCB.

Catatan :

Perlu ditinjau kembali untuk ketebalan PCB yang ideal.

f. Hasil pemasangan komponen pada PCB



Gambar 17. Hasil Pemasangan Komponen Pada PCB

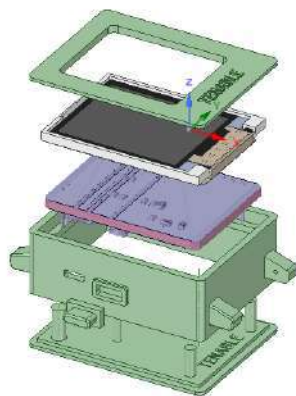
Pemasangan komponen pada PCB menggunakan dua cara penyolderan yaitu dengan solder biasa dan blower (solder uap). Untuk penggunaan solder uap suhu diatur pada 350° dan tekanan angin seminim mungkin. Penggunaan flux sangat disarankan agar panas pada timah tersebar secara merata dan hasilnya maksimal.

Catatan :

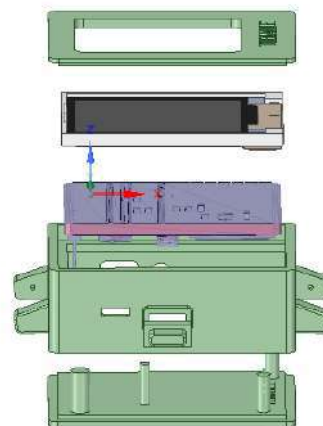
Dalam proses menyolder sebuah komponen SMD menggunakan solder uap terkadang mempengaruhi komponen lain yang sudah terpasang. Hal tersebut bisa saja membuat komponen SMD lain terlepas, melepuh atau tidak terpasang benar pada kaki-kakinya. Oleh karena itu kedepannya diperlukan sebuah isolasi lem anti panas yang dipasangkan pada komponen disekitar komponen SMD yang akan dipasang atau dilepas.

3. Desain case

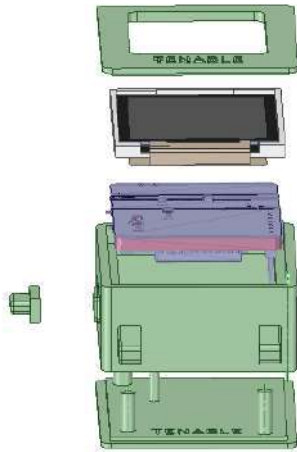
a. Desain case jam tangan TENABLE



Gambar 18. Desain Case Jam Tangan Isometri

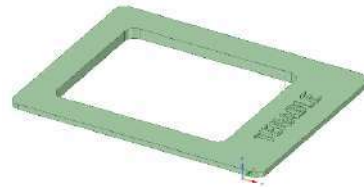


Gambar 19. Desain Case Jam Tangan Samping



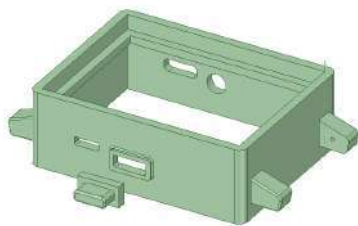
Gambar 20. Desain Case Jam

Tangan Depan



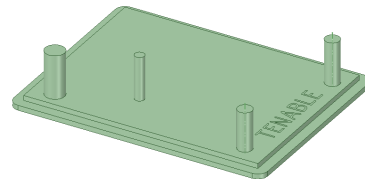
Gambar 21. Desain Cover LCD Jam

Tangan



Gambar 22. Desain Body Jam

Tangan



Gambar 23. Desain Penutup Bawah

Jam Tangan

Telah disinggung sebelumnya pada desain pcb, bahwa desain case jam tangan memiliki bentuk yang boxy karena desainnya yang mengikuti bentuk PCB.

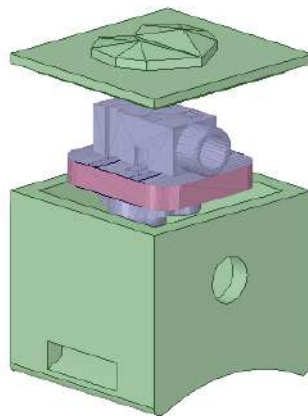
Catatan :

- Desain case jam tangan masih terlalu boxy dan kurang fashionable.
- Selain itu desain case jam tangan ini memiliki kekurangan karena masih kurang memperhatikan aspek mekanikal seperti tombol, switch on/off, port sensor dan port charging. Hal tersebut menyebabkan komponen pada PCB rentan lepas sampai permukaan untuk

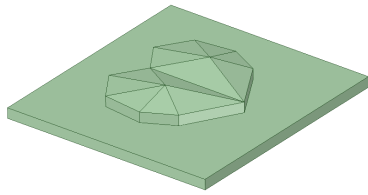
soldernya pun lepas sehingga tidak bisa dipasang kembali. Terlepasnya komponen tersebut dialami oleh device TENABLE V1 yang mana komponen port charging dan tombol tidak dapat dipasang kembali.

- Tidak adanya jarak (gap) antara penutup case jam tangan baik itu penutup lcd dan penutup bagian bawah dengan lubangnya menyebabkan sulitnya pemasangan penutup lcd dan penutup bagian bawah. Memberikan jarak antara ukuran penutup dan lubangnya sebesar 1mm adalah solusinya.
- Menambahkan sistem pengunci secara mekanikal pada penutup case jam tangan dapat menghindari penggunaan lem yang dirasa kurang fleksibel.

b. Desain case cincin TENABLE

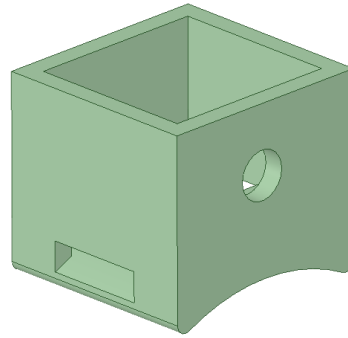


Gambar 24. Desain Case Cincin Isometri



Gambar 25.

Desain Tutup Cincin



Gambar 26.

Desain Body Cincin

Desain case cincin TENABLE sama-sama memiliki bentuk yang boxy dengan jam tangannya. Alasan mengapa bentuknya seperti ini juga sama dikarenakan desain cincin mengikuti bentuk pcb.

Catatan :

- Desain cincin dirasa masih kurang dari segi tampilannya dan kenyamanannya saat digunakan.
- Tidak adanya jarak (gap) antara ukuran penutup case cincin dan lubangnya menyebabkan penutup cincin susah untuk disematkan. Memberikan jarak antara ukuran penutup cincin dan lubangnya sebesar 1mm adalah solusinya.
- Menambahkan sistem pengunci secara mekanikal pada penutup case cincin dapat menghindari penggunaan lem yang dirasa kurang fleksibel.

c. Hasil print 3D case jam tangan dan cincin



Gambar 27. Hasil Cetak Print 3D Case Jam Tangan dan Cincin

Case jam tangan TENABLE menggunakan warna kuning dan case cincinnya menggunakan warna hitam. Pemilihan warna kuning dan hitam dirasa cocok dan sedikit menutupi kekurangan desainnya yang masih perlu diperbaiki lagi.

Catatan :

Case TENABLE pada saat selesai dicetak memiliki permukaan yang masih kasar terutama pada beberapa lekukan-lekukan case dan lubang case seperti untuk port charging yang kurang rapih. Sehingga perlu kami rapihkan dan menguras waktu. Disini memilih jasa print 3D yang dapat handle proses sampai finishing menjadi solusinya. Dengan demikian pengembang dapat lebih fokus pada hal lain yang lebih penting dalam menyempurnakan device TENABLE ini.

4. Hasil Akhir



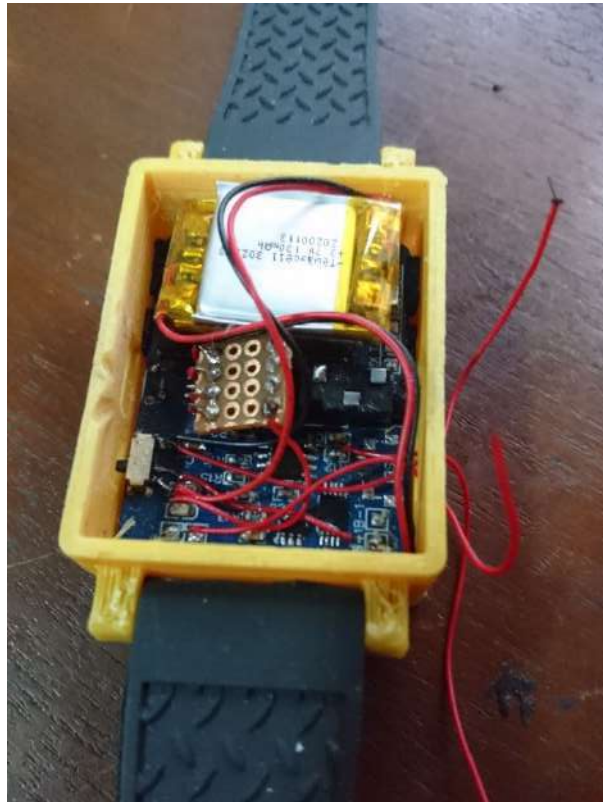
Gambar 28. Case Jam Tangan dan Cincin Setelah Finishing



Gambar 29. Hasil Perakitan Jam Tangan



Gambar 30. Hasil Perakitan Jam Tangan dan Cincin yang Disambungkan



Gambar 31. Hasil Akhir PCB TENABLE V1

Setelah proses perakitan dan berbagai trouble shoot ada beberapa perubahan pada rangkaian PCB yang diakali dengan penyambungan jumper (kabel). Perubahan ini berkaitan dengan rangkaian PCB yang tidak sesuai yang telah disinggung pada skematik rangkaian diatas.



Gambar 32. Hasil Perakitan Cincin



Gambar 33. Hasil Akhir PCB Cincin

5. Program device TENABLE

a. Isi Program

```
//Tambahkan Library Disini
//*****

//Template Coding Arduino OTA on ESP32 (JANGAN
DIHAPUS)
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>

#define TFT_CS      5
#define TFT_RST    4
#define TFT_DC      2

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,
TFT_DC, TFT_RST);

const char* ssid = "NamaWiFi";
const char* password = "PasswordWiFi";

//Inisialisasi Variabel Disini
//*****

#define length 60
```



```

int temp[length];
float alpha = 0.8;
double current_output, previous_output;
int current_input, afterFilter;
unsigned long previousMillis = 0;
const long interval = 1000;
int pinBacklight = 33;
int pinButton = 32;
int buttonState = 0;
bool pressed = false;
bool lcd = true;

//Custom Function Disini
//*****
int movAvg(int in) {
    double sum = 0;
    for (int i = length - 1; i >= 1; i--) {
        temp[i] = temp[i - 1];
        sum += temp[i];
    }
    temp[0] = in;
    sum += in;
    return (sum / length);
}

void setup() {
    // Setup Arduino OTA (JANGAN DIHAPUS)
    pinMode(pinBacklight, OUTPUT);
    digitalWrite(pinBacklight, HIGH);
    tft.initR(INITR_BLACKTAB); // initialize a
ST7735S chip, black tab
    tft.fillScreen(ST77XX_WHITE);
    tft.setTextWrap(false);

    tft.println("Booting");
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    while (WiFi.waitForConnectResult() !=
WL_CONNECTED) {
        Serial.println("Connection Failed!
Rebooting...");
        tft.println("Connection Failed! Rebooting...");
        delay(5000);
        ESP.restart();
    }

    // Port defaults to 3232
    // ArduinoOTA.setPort(3232);

    // Hostname defaults to esp3232-[MAC]

```

```

// ArduinoOTA.setHostname("myesp32");

// No authentication by default
// ArduinoOTA.setPassword("admin");

// Password can be set with it's md5 value as well
// MD5(admin) = 21232f297a57a5a743894a0e4a801fc3
//
ArduinoOTA.setPasswordHash("21232f297a57a5a743894a0e4a801fc3");

ArduinoOTA
.onStart([]() {
    String type;
    if (ArduinoOTA.getCommand() == U_FLASH)
        type = "sketch";
    else // U_SPIFFS
        type = "filesystem";

    // NOTE: if updating SPIFFS this would be the
place to unmount SPIFFS using SPIFFS.end()
    Serial.println("Start updating " + type);
})
.onEnd([]() {
    Serial.println("\nEnd");
    tft.println("\nEnd");
})
.onProgress([](unsigned int progress, unsigned
int total) {
    Serial.printf("Progress: %u%%\r", (progress /
(total / 100)));
    tft.printf("Progress: %u%%\r", (progress /
(total / 100)));
})
.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    tft.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) {
        Serial.println("Auth Failed");
        tft.println("Auth Failed");
    }
    else if (error == OTA_BEGIN_ERROR) {
        Serial.println("Begin Failed");
        tft.println("Begin Failed");
    }
    else if (error == OTA_CONNECT_ERROR) {
        Serial.println("Connect Failed");
        tft.println("Connect Failed");
    }
    else if (error == OTA_RECEIVE_ERROR) {

```

```

        Serial.println("Receive Failed");
        tft.println("Receive Failed");
    }
    else if (error == OTA_END_ERROR) {
        Serial.println("End Failed");
        tft.println("End Failed");
    }
});

ArduinoOTA.begin();

Serial.println("Ready");
tft.println("Ready");
Serial.print("IP address: ");
tft.println("IP address: ");
Serial.println(WiFi.localIP());
tft.println(WiFi.localIP());
delay(3000);
//  Setup Program nya Disini
//*****
pinMode(pinButton, INPUT_PULLUP);
tft.setCursor(25, 130);
tft.setTextColor(ST77XX_BLUE);
tft.setTextSize(2);
tft.print("TENABLE");
tft.setTextSize(1);
tft.setCursor(50, 118);
tft.print("V 0.1");
tft.setCursor(0, 0);
tft.setTextSize(1);
tft.setTextColor(ST77XX_WHITE);
}

void loop() {
//  Loop Arduino OTA (JANGAN DIHAPUS)
    ArduinoOTA.handle();

//  Loop Program disini
//*****
//  put your main code here, to run repeatedly:
//  Membaca data analog mentah
    int current_input = analogRead(35);

//Low Pas Filter
    current_output = alpha * current_input + (1 -
alpha) * previous_output;
    previous_output = current_output;

//Moving Average
    afterFilter = movAvg(current_output);

```

```

    if (currentMillis - previousMillis >= interval) {
        previousMillis1 = currentMillis;
        tft.fillRoundRect(10, 5, 100, 15, 10,
ST77XX_RED);
        tft.setCursor(15, 10);
        tft.print("Input ADC ");
        tft.println(current_input);
        tft.fillRoundRect(10, 25, 100, 15, 10,
ST77XX_RED);
        tft.setCursor(15, 30);
        tft.print("LPF ");
        tft.println(current_output);
        tft.fillRoundRect(10, 45, 100, 15, 10,
ST77XX_RED);
        tft.setCursor(15, 50);
        tft.print("Filter ");
        tft.println(afterFilter);
    }

    buttonState = digitalRead(pinButton);
    if (buttonState == LOW && lcd == true && pressed
== false) {
        digitalWrite(pinBacklight, LOW);
        lcd = false;
        pressed = true;
    } else if (buttonState == LOW && lcd == false &&
pressed == false) {
        digitalWrite(pinBacklight, HIGH);
        lcd = true;
        pressed = true;
    } else if (buttonState == HIGH && pressed == true)
{
        pressed = false;
    }
}

```

b. Penjelasan Program

i. Template Program OTA untuk ESP32

```

//Tambahkan Library Disini
//*****

//Template Coding Arduino OTA on ESP32 (JANGAN
DIHAPUS)
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

```

```

#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>

#define TFT_CS      5
#define TFT_RST     4
#define TFT_DC      2

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,
TFT_DC, TFT_RST);

const char* ssid = "NamaWiFi";
const char* password = "PasswordWiFi";

//Inisialisasi Variabel Disini
//*****

void setup() {
//  Setup Arduino OTA (JANGAN DIHAPUS)
  pinMode(33, OUTPUT);
  digitalWrite(33, HIGH);
  tft.initR(INITR_BLACKTAB); // initialize a
ST7735S chip, black tab
  tft.fillScreen(ST77XX_BLACK);
  tft.setTextWrap(false);
  tft.setCursor(0, 0);
  tft.setTextColor(ST77XX_WHITE);

  Serial.begin(115200);
  Serial.println("Booting");
  tft.println("Booting");
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.waitForConnectResult() !=
WL_CONNECTED) {
    Serial.println("Connection Failed!
Rebooting...");
    tft.println("Connection Failed!
Rebooting...");
    delay(5000);
    ESP.restart();
  }

  // Port defaults to 3232
  // ArduinoOTA.setPort(3232);

  // Hostname defaults to esp3232-[MAC]
  // ArduinoOTA.setHostname("myesp32");

```

```

// No authentication by default
// ArduinoOTA.setPassword("admin");

// Password can be set with it's md5 value as
well
// MD5(admin) =
21232f297a57a5a743894a0e4a801fc3
//
ArduinoOTA.setPasswordHash("21232f297a57a5a743894
a0e4a801fc3");

ArduinoOTA
.onStart([]() {
    String type;
    if (ArduinoOTA.getCommand() == U_FLASH)
        type = "sketch";
    else // U_SPIFFS
        type = "filesystem";

    // NOTE: if updating SPIFFS this would be
the place to unmount SPIFFS using SPIFFS.end()
    Serial.println("Start updating " + type);
})
.onEnd([]() {
    Serial.println("\nEnd");
    tft.println("\nEnd");
})
.onProgress([](unsigned int progress,
unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress
/ (total / 100)));
    tft.printf("Progress: %u%%\r", (progress /
(total / 100)));
})
.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    tft.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) {
        Serial.println("Auth Failed");
        tft.println("Auth Failed");
    }
    else if (error == OTA_BEGIN_ERROR) {
        Serial.println("Begin Failed");
        tft.println("Begin Failed");
    }
    else if (error == OTA_CONNECT_ERROR) {
        Serial.println("Connect Failed");
        tft.println("Connect Failed");
    }
    else if (error == OTA_RECEIVE_ERROR) {

```

```

        Serial.println("Receive Failed");
        tft.println("Receive Failed");
    }
    else if (error == OTA_END_ERROR) {
        Serial.println("End Failed");
        tft.println("End Failed");
    }
}
});

ArduinoOTA.begin();

Serial.println("Ready");
tft.println("Ready");
Serial.print("IP address: ");
tft.println("IP address: ");
Serial.println(WiFi.localIP());
tft.println(WiFi.localIP());
tft.println("Berhasil");

//  Setup Program nya Disini
//*****

}

void loop() {
//  Loop Arduino OTA (JANGAN DIHAPUS)
    ArduinoOTA.handle();

//  Loop Program disini
//*****

}

```

Program OTA (On The Air) ESP32 dapat diakses dari Example ESP32. Program OTA ini berguna agar programmer dapat memprogram device dengan microcontroller ESP32 secara wireless menggunakan jaringan WiFi. Serangkaian baris kodenya harus dicantumkan atau digabungkan dengan program utama yang akan programmer upload. Apabila serangkaian kode OTA tidak dicantumkan kembali pada saat memprogram device maka fitur OTA tidak berfungsi pada program yang baru.

Ada modifikasi pada kode program OTA yang digunakan oleh device TENABLE ini yaitu mengganti fungsi print text yang sebelumnya dilakukan pada serial monitor sebagai indikator sistem OTA menjadi fungsi print text pada layar LCD TFT yang digunakan pada device TENABLE.

ii. Import Library

```
//Library Wifi dan ESP32
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>

//Library Arduino OTA
#include <ArduinoOTA.h>

//Library LCD
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>
```

Awal program dimulai dengan memanggil beberapa library diantaranya library WiFi, ESP32, Arduino OTA dan LCD

iii. Inisialisasi Variabel

- Pin LCD

```
//Inisialisasi Pin LCD untuk komunikasi SPI
dengan esp32
#define TFT_CS      5
#define TFT_RST     4
#define TFT_DC      2

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,
TFT_DC, TFT_RST);
```

Inisialisasi pin LCD digunakan agar ESP32 dapat menampilkan pesan di LCD melalui komunikasi SPI.

- Konfigurasi WiFi

```
const char* ssid = "NamaWiFi";
const char* password = "PasswordWiFi";
```


Inisialisasi variabel ssid dan password yang akan disambungkan.

- Variabel Untuk Filter Moving Average

```
#define length 60 //Panjang dari char/array
int temp[length]; //char dengan nama variabel
temp dengan panjangnya sebesar length yaitu 60
```

Dalam melakukan filter dengan Moving Average dibutuhkan sebuah temporary variabel dalam pengoperasiannya

- Variabel Untuk Low Pass Filter

```
float alpha = 0.8;
double current_output, previous_output;
```

- Variabel Untuk Pembacaan Sensor Analog dan Hasil Filter Moving Average

```
int current_input, afterFilter;
```

Variabel current_input menampung nilai analog sensor yang dibaca sedangkan variabel afterFilter menampung nilai hasil akhir dari filter Moving Average.

- Variabel Untuk Metode Millis

```
unsigned long previousMillis = 0;
const long interval = 1000;
```

Penggunaan millis bertujuan agar process jeda dilakukan tanpa fungsi delay(), yang mana fungsi delay() ini kurang baik untuk kinerja controller.

- Pin Backlight LCD

```
int pinBacklight = 33;
bool lcd = true;
```

Backlight pada LCD dapat diatur hidup dan matinya dengan cara menyambungkan IO33 dengan pin LED LCD. Pengaturan hidup dan mati backlight LCD bertujuan agar penggunaan baterai menjadi lebih hemat. Variabel pinBacklight menampung nilai pin berapa ESP32 yang tersambung dengan pin LED LCD, dan

variabel lcd menampung status dari backlight LCD apakah nyala atau mati, true itu nyala dan false itu mati.

- Pin Pushbutton

```
int pinButton = 32;
int buttonState = 0;
bool pressed = false;
```

Pushbutton dihubungkan dengan IO32 digunakan untuk switching nyala dan mati backlight LCD. Variabel pinButton menampung nilai pin berapa yang digunakan push button, variabel buttonState menampung nilai toggle nyala dan mati atau 0 dan 1, sedangkan variabel pressed menampung kondisi apakah push button sedang ditekan atau tidak.

iv. Function

- Moving Average

```
int movAvg(int in) {
    double sum = 0;
    for (int i = length - 1; i >= 1; i--) {
        temp[i] = temp[i - 1];
        sum += temp[i];
    }
    temp[0] = in;
    sum += in;
    return (sum / length);
}
```

Fungsi moving average digunakan untuk memfilter sinyal. Fungsi ini dipanggil didalam program secara loop.

v. Void Setup

Pada fungsi void setup() program didalamnya dieksekusi diawal saat device baru dinyalakan. Program-program yang harus dijalankan saat device pertama kali boot diantaranya sebagai berikut :

- Menyalakan backlight LCD pada saat device dinyalakan

```
pinMode(pinBacklight, OUTPUT);
digitalWrite(pinBacklight, HIGH);
```

- Setup tampilan LCD

```
tft.initR(INITR_BLACKTAB);
tft.fillScreen(ST77XX_WHITE); //setting
background dengan warna putih
tft.setTextWrap(false); //disable text wrap
```

- Menghubungkan ke jaringan WiFi

```
tft.println("Booting");
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
while (WiFi.waitForConnectResult() !=
WL_CONNECTED) {
    Serial.println("Connection Failed!
Rebooting...");
    tft.println("Connection Failed!
Rebooting...");
    delay(5000);
    ESP.restart();
}
```

- Setting IO32 yang digunakan push button dengan Internal Resistor PULL UP

```
pinMode(pinButton, INPUT_PULLUP);
```

Penggunaan Internal Resistor PULL UP karena push button disambungkan langsung ke ground dan ke pin 32 tanpa menggunakan external resistor.

- Menampilkan label TENABLE pada LCD

```
tft.setCursor(25, 130);
tft.setTextColor(ST77XX_BLUE);
tft.setTextSize(2);
tft.print("TENABLE");
tft.setTextSize(1);
tft.setCursor(50, 118);
tft.print("V 1.0");
```

```
tft.setCursor(0, 0);
tft.setTextSize(1);
tft.setTextColor(ST77XX_WHITE);
```

vi. Void Loop()

- Membaca sinyal analog dari sensor

```
// Membaca data analog mentah
int current_input = analogRead(35);
```

- Low Pass Filter

```
//Low Pas Filter
current_output = alpha * current_input + (1
- alpha) * previous_output;
previous_output = current_output;
```

- Moving Average

```
//Moving Average
afterFilter = movAvg(current_output);
```

- Menampilkan nilai mentah dari pembacaan analog, low pass filter dan moving average setiap 1 detik sekali.

```
if (currentMillis - previousMillis >=
interval) {
    previousMillis1 = currentMillis;
    tft.fillRoundRect(10, 5, 100, 15, 10,
ST77XX_RED);
    tft.setCursor(15, 10);
    tft.print("Input ADC ");
    tft.println(current_input);
    tft.fillRoundRect(10, 25, 100, 15, 10,
ST77XX_RED);
    tft.setCursor(15, 30);
    tft.print("LPF ");
    tft.println(current_output);
    tft.fillRoundRect(10, 45, 100, 15, 10,
ST77XX_RED);
    tft.setCursor(15, 50);
    tft.print("Filter ");
    tft.println(afterFilter);
}
```

- Menyalakan dan mematikan backlight LCD dengan push button

```

buttonState = digitalRead(pinButton);
if (buttonState == LOW && lcd == true &&
pressed == false){
    digitalWrite(pinBacklight, LOW);
    lcd = false;
    pressed = true;
} else if (buttonState == LOW && lcd ==
false && pressed == false) {
    digitalWrite(pinBacklight, HIGH);
    lcd = true;
    pressed = true;
} else if (buttonState == HIGH && pressed ==
true) {
    pressed = false;
}
}

```

vii. Catatan

Program yang digunakan pada device TENABLE V1 masih memiliki kekurangan dalam mengefisiensikan penggunaan daya. Salah satunya adalah mode Modem Sleep pada ESP32 yang tidak disematkan pada program saat ini. Modem Sleep ini merupakan kondisi dimana koneksi WiFi dimatikan pada saat tertentu dan akan diaktifkan pada saat yang diinginkan. Hal ini dapat diimplementasikan pada device TENABLE dengan WiFi yang dimatikan saat device sedang tidak dalam kondisi mengirimkan data log, layar LCD dinyalakan dan notifikasi adanya peningkatan kadar gula darah. Sehingga kedepannya device TENABLE memiliki konsumsi daya baterai lebih efisien.

6. Testing





Langkah terakhir dari pengerjaan alat ini adalah pengujian. Untuk device TENABLE V1 pada pembahasan di laporan ini hanya berfokus pada pengujian nilai sinyal yang dibaca dalam bentuk ADC (Analog Digital Converter) bukan dalam bentuk satuan nilai kandungan gula darah. Berikut ini dokumentasi pengujian yang telah dilakukan :




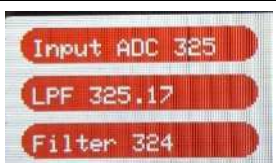








Gambar 34. Pengujian saat cincin TENABLE dibiarkan terbuka



Gambar 35. Pengujian saat cincin TENABLE ditempelkan ke jari

Penggunaan Cincin	Nilai ADC yang terukur
	
	

Tabel 3. Pengujian nilai ADC saat cincin digunakan dan tidak.

Dari gambar 34 & 35 dan tabel 3, dapat disimpulkan bahwa terjadi perubahan nilai ADC saat cincin digunakan dan tidak. Saat cincin TENABLE dibiarkan terbuka maka nilai ADC yang terukur sekitar 200 – 250, sedangkan saat cincin TENABLE ditempelkan ke jari layaknya cincin sedang dipakai maka nilai ADC yang terukur mencapai 300 lebih bahkan hampir menyentuh 400. Tentunya nilai ini akan berbeda pada pengujian dengan orang yang berbeda-beda.

Pengaruh cahaya dari luar dan penguatan sinyal membuat nilai ADC dalam kondisi sensor tidak menerima pantulan dari infrared sebesar 200, dan bukan 0. Maka jika dirasa perlu untuk kedepannya dapat kalibrasi agar kondisi idle cincin tanpa digunakan memiliki nilai ADC 0.

Saat jari ditempelkan pada cincin TENABLE terjadi lonjakan nilai ADC yang menandakan bahwa cahaya infrared yang dipancarkan dapat diterima oleh sensor fotodiode. Dengan begitu sudah bisa dipastikan bahwa cahaya infrared yang dipancarkan memantul kembali, hal ini sesuai dengan konsep awal bahwa cahaya infrared yang menembus kulit dan melewati pembuluh darah kemudian dipantulkan

oleh tulang. Ada satu hal yang mesti ditinjau lebih pada pengembangan TENABLE selanjutnya, yaitu pengaruh figmen kulit terhadap nilai ADC yang terukur.

Agar device TENABLE dapat menampilkan nilai kandungan gula darah diperlukan pengumpulan data nilai kandungan gula darah dan nilai ADC nya dengan mengukur sekian banyak sampel orang. Setelah data didapat maka persamaan akan diperoleh sehingga dapat kita masukan sebagai konversi nilai ADC ke kandungan gula darah. Namun hal tersebut tidak dikerjakan dan dibahas pada laporan projek ini.

C. Kesimpulan

Pengerjaan projek device TENABLE V1 tentu masih jauh dari kata sempurna, banyak hal yang harus di improve kedepannya sampai device ini layak untuk disebar luaskan dan dirasakan manfaatnya. Oleh karena itu penulis telah menyematkan catatan per subpoint nya pada point pembahasan. Catatan tersebut dapat menjadi acuan untuk pengembang selanjutnya meningkatkan dan menyelesaikan masalah-masalah pada device TENABLE V1. Meskipun masih memiliki banyak kekurangan dan belum maksimal namun device TENABLE V1 pada projek ini telah memenuhi goals-goals yang telah ditentukan.

D. Dokumentasi Kerja



