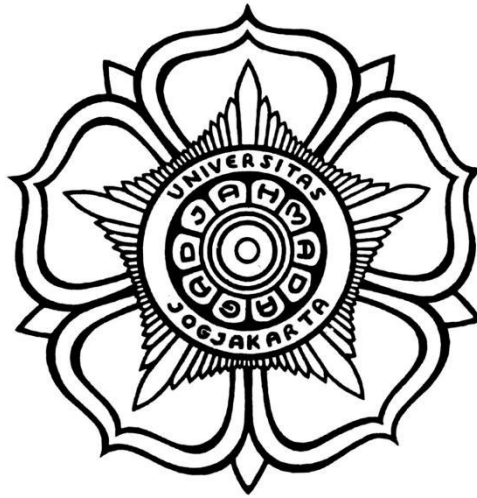


PERANCANGAN RANGKAIAN, PROGRAM DAN SIMULASI PADA MAPLE MINI

**LAPORAN TUGAS TENGAH SEMESTER
SISTEM BERDASAR MIKROPROSESOR
(TKEE163222)**



Disusun Oleh:

Faishal Reza Adhitama
16/395393/TK/44685

Raka Andinan Pratama
16/395856/TK/44738

Resha Dwika Hefni Al-Fahsi
16/394959/TK/44251

Vicko Pranowo
16/394966/TK/44258

**DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
2019**

Abstrak

Laporan ini membahas tentang perancangan rangkaian, program dan simulasi pada Maple Mini. Maple Mini adalah jenis yang kedua dari dua papan STM32F103 yang dibuat oleh LeafLabs. Maple Mini asli dirilis pada tahun 2011 dan saat ini tidak lagi tersedia, tetapi beberapa perusahaan membuat tiruan, yang memiliki fungsi yang hampir identik. Dalam perancangan firmware yang digunakan pada Maple Mini digunakan Arduino IDE sebagai perantara bahasa pemrograman dengan bahasa mesin. Arduino¹ IDE digunakan untuk proses *flash* yaitu memasukan program ke dalam IC STM32F103 pada Maple Mini. Seluruh program yang dituliskan pada laporan ini tersedia online pada <https://github.com/fareadh/sbm2k19-sub2-genap>.

Copyright © 2019 Faishal Reza Adhitama, Raka Andinan Pratama, Resha Dwika Hefni Al-Fahsi, Vicko Pranowo from Universitas Gadjah Mada.

This document may be used for academic purposes with reference to this document.

¹Arduino is trademark of Arduino LLC USA.

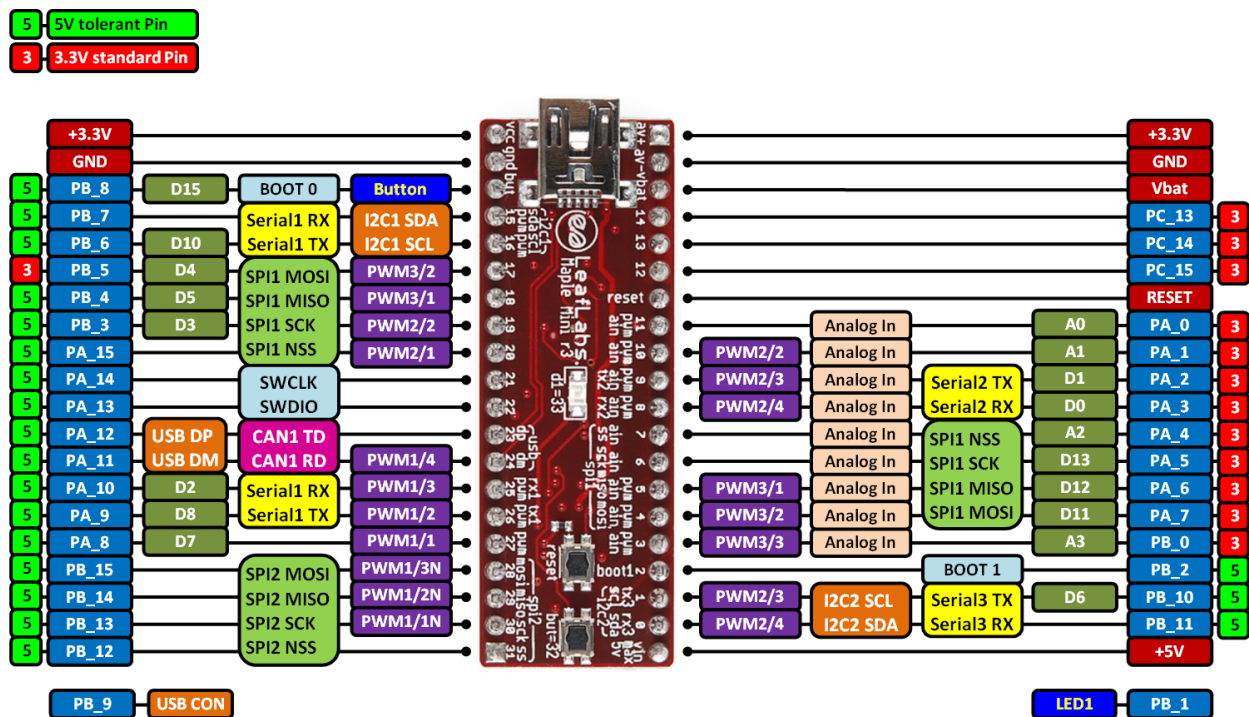
Daftar Isi

Abstrak	1
Daftar Isi	2
Bab 1: Pendahuluan	3
Bab 2: Blink LED	4
2.1. Rangkaian	4
2.2. Program	4
Bab 3: Keypad	5
3.1. Rangkaian	5
3.2. Program	6
3.3. Simulasi	8
Bab 4: Joystick	15
4.1. Rangkaian	15
4.2. Program	16
4.3. Simulasi	17
Bab 5: Realtime Clock	18
5.1. Rangkaian	18
5.2. Program	19
5.3. Simulasi	21
Daftar Pustaka	22
A. Setup Maple Mini pada Arduino IDE	23

Bab 1

Pendahuluan

Tujuan dari pembuatan laporan ini untuk memberikan penjelasan tentang perancangan rangkaian, program dan simulasi pada Maple Mini. Board Maple Mini menggunakan STM32F103 sebagai chip IC mikrokontrolernya. Untuk dapat mengakses dan memanipulasi pinout yang terdapat pada Maple Mini (yang ditunjukkan pada gambar) dilakukan *flash firmware* ke dalam IC mikrokontroler Maple Mini. Proses tersebut dilakukan dengan melakukan kompilasi program pada Arduino IDE kemudian dilakukanlah proses *flash* tersebut.



Gambar 1. Skematik Pinout pada Maple Mini.

Pada bab 2, kita melakukan perancangan program Blink LED pada Maple Mini. Pada Maple Mini terdapat LED internal yang dapat diakses melalui program dan dapat dibuat berkedip (*blink*).

Pada bab 3 dilakukan perancangan program, simulasi dan rangkaian keypad pada Maple Mini. Keypad tersebut terdiri atas 16 tombol dan 8 pin yang dibaca dengan isyarat digital.

Pada bab 4 selain keypad dapat juga melakukan perancangan program, simulasi dan rangkaian joystick pada Maple Mini. Joystick memiliki 5 buah pin yang terdiri atas V_{CC}, Ground dan 3 pin output digital berupa data X, data Y dan Switch.

Pada bab 5 dibahas mengenai perancangan program, simulasi dan rangkaian Realtime Clock pada Maple Mini. Realtime Clock memiliki protokol komunikasi berupa I²C sehingga terdapat 4 pin yang terdiri atas SDA, SCL, V_{CC} dan Ground.

Bab 2

Blink LED

LED adalah suatu semikonduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju. Gejala tersebut merupakan bentuk elektroluminesensi. Warna yang dihasilkan bergantung pada bahan semikonduktor yang dipakai. LED dibuat pertama kali oleh Shuji Nakamura pada tahun 1993. Pada saat itu beliau bekerja di Nichia Corporation di Jepang.

2.1. Rangkaian

Pada Maple Mini terdapat internal LED pada pin 33. Secara fisik pin 33 tidak berbentuk pinout, namun khusus untuk LED internal pada Maple Mini. Sehingga untuk rangkaian yang digunakan pada Blink LED tidak memerlukan LED, kabel dan *breadboard* tambahan. Blink LED sendiri dilakukan dengan menyalakan dan mematikan LED pada rentang waktu tertentu.

2.2. Program

Program yang akan dirancang akan menyalakan dan mematikan LED internal pada Maple Mini selama 1000 ms atau satu detik. Sehingga LED akan menyala selama satu detik kemudian mati selama satu detik pula. Digunakan alias LED_BUILTIN untuk pin 33 pada Maple Mini. Pada Arduino IDE dapat dituliskan²:

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

² Selengkapnya: <https://github.com/fareadh/sbm2k19-sub2-genap/blob/master/blinkled/blinkled.ino>.

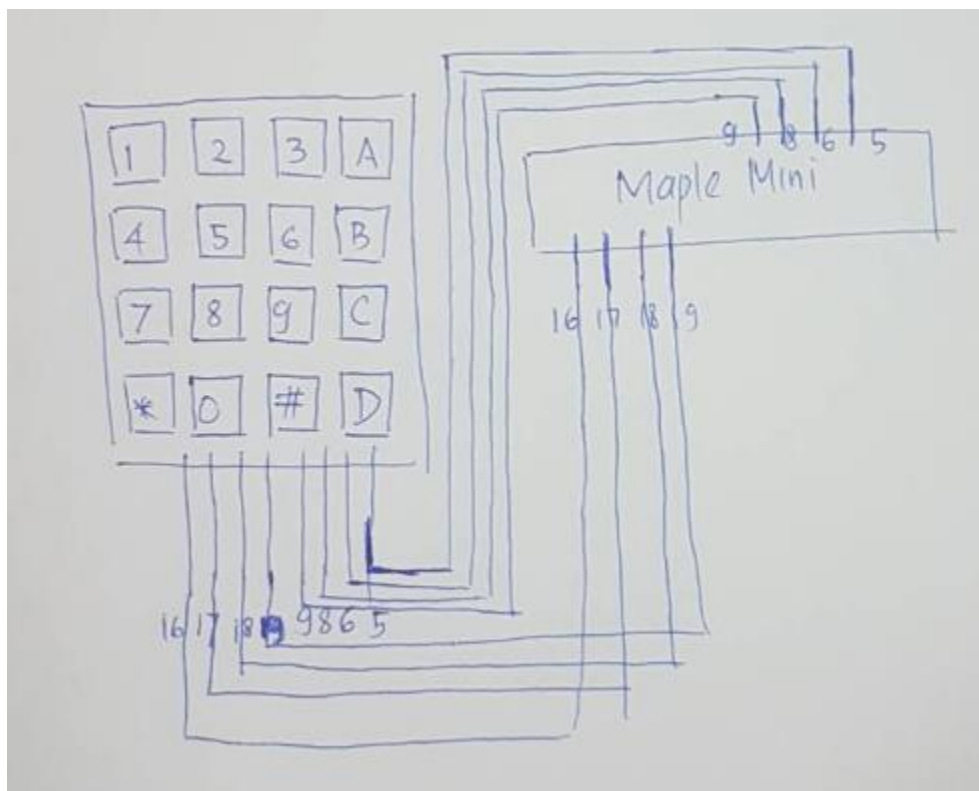
Bab 3

Keypad

Keypad merupakan seperangkat tombol untuk operasi portabel perangkat elektronik, telepon, atau peralatan lainnya. Keypad merupakan sebuah rangkaian tombol yang tersusun atau dapat disebut *pad* yang biasanya terdiri dari huruf, simbol dan angka. Keypad ditemukan pertama kalinya oleh John Elias Karlin yang lahir pada Februari 28, 1918, di Johannesburg. Karlin merupakan seorang psikolog industri di Bell Labs di Murray Hill.

3.1. Rangkaian

Keypad yang digunakan untuk Maple Mini pada laporan ini berupa keypad 4x4. Tombol pada keypad tersebut terdiri atas angka (0-9), huruf (A, B, C, D) dan symbol (* dan #). Keypad tersebut memiliki 8 buah pin digital yang terdiri atas ouput pada baris dan input pada kolom. Pin yang digunakan untuk input kolom yaitu pin 9, pin 8, pin 6 dan pin 5. Pin yang digunakan sebagai output baris yaitu pin 16, pin 17, pin 18 dan pin 19.



Gambar 2. Rangkaian Keypad pada Maple Mini.

3.2. Program

Dengan memanipulasi pin output baris dengan memberikan isyarat digital dari Maple Mini dan pin input baris dengan membaca isyarat digital yang dihasilkan setelah menekan suatu tombol pada keypad dapat ditentukan tombol yang sedang ditekan. Pada Arduino IDE dapat dituliskan³:

```
byte h=0, v=0;
const unsigned long period=50;
unsigned long kdelay=0;
const byte rows=4;
const byte columns=4;
const byte Output[rows] = {16,17,18,19};
const byte Input[columns] = {9,8,6,5};

byte keypad()
{
    static bool no_press_flag=0;
    for(byte x=0;x<columns;x++)
    {
        if (digitalRead(Input[x])==HIGH);
        else
            break;
        if(x==(columns-1))
        {
            no_press_flag=1;
            h=0;
            v=0;
        }
    }
    if(no_press_flag==1)
    {
        for(byte r=0;r<rows;r++)
            digitalWrite(Output[r],LOW);
        for(h=0;h<columns;h++)
        {
            if(digitalRead(Input[h])==HIGH)
                continue;
            else
            {
                for (v=0;v<rows;v++)
                {
                    digitalWrite(Output[v],HIGH);
                    if(digitalRead(Input[h])==HIGH)
                    {
                        no_press_flag=0;
                        for(byte w=0;w<rows;w++)
                            digitalWrite(Output[w],LOW);
                        return v*4+h;
                    }
                }
            }
        }
    }
    return 50;
}
```

```

void setup()
{
  for(byte i=0;i<rows;i++)
  {
    pinMode(Output[i],OUTPUT);
  }
  for(byte s=0;s<columns;s++)
  {
    pinMode(Input[s],INPUT_PULLUP);
  }
  Serial.begin(9600);
}
void loop()
{
  if(millis()-kdelay>period)
  {
    kdelay=millis();
    switch (keypad())
    {
      case 0:
        Serial.println(1);
        break;
      case 1:
        Serial.println(2);
        break;
      case 2:
        Serial.println(3);
        break;
      case 3:
        Serial.println("A");
        break;
      case 4:
        Serial.println(4);
        break;
      case 5:
        Serial.println(5);
        break;
      case 6:
        Serial.println(6);
        break;
      case 7:
        Serial.println("B");
        break;
      case 8:
        Serial.println(7);
        break;
      case 9:
        Serial.println(8);
        break;
      case 10:
        Serial.println(9);
        break;
      case 11:
        Serial.println("C");
        break;
    }
  }
}

```



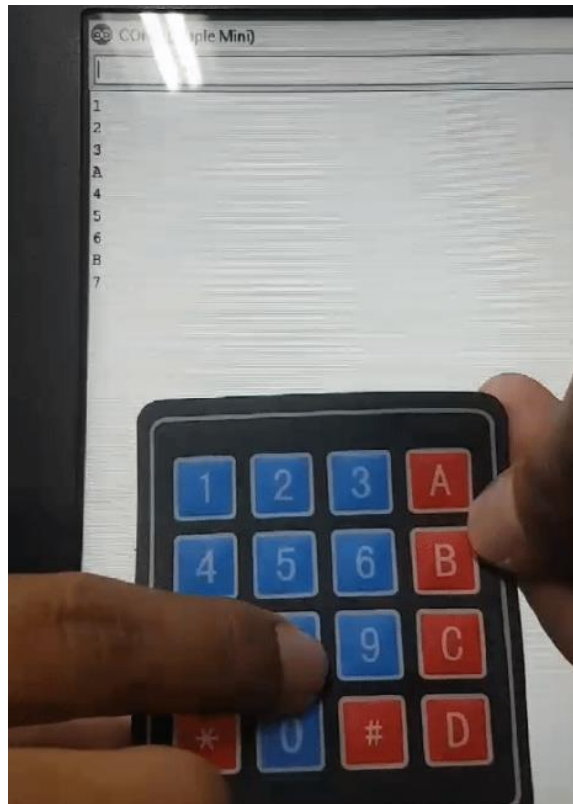
```

        case 12:
            Serial.println("*");
        break;
        case 13:
            Serial.println(0);
        break;
        case 14:
            Serial.println("#");
        break;
        case 15:
            Serial.println("D");
        break;
        default:
            ;
    }
}

```

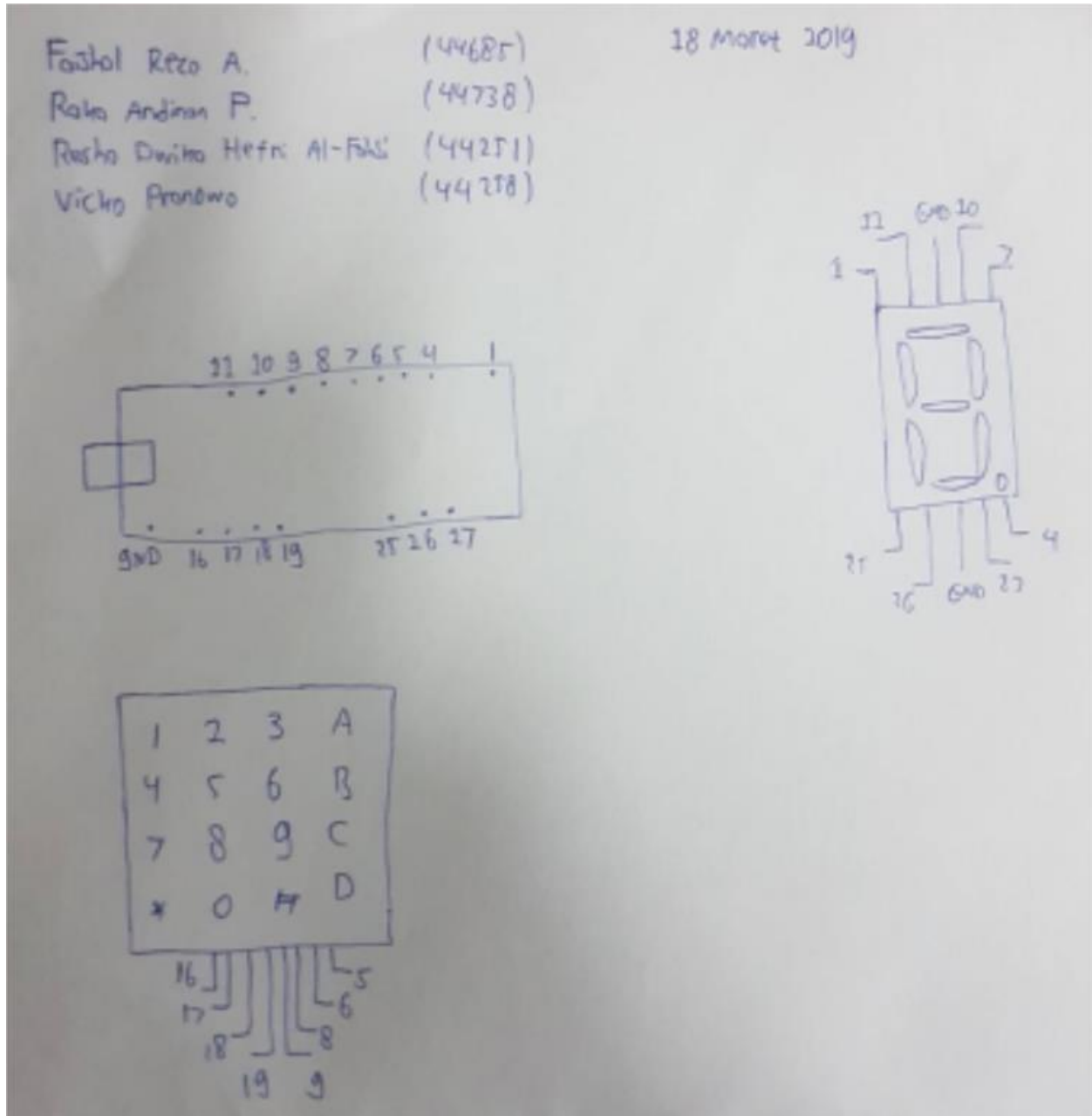
3.3. Simulasi

Kemudian hasil perancangan rangkaian dan program tersebut diimplementasikan pada Maple Mini. Kemudian keypad disimulasikan dengan menekan tombol pada keypad dan mencetak karakter yang ditekan ke layar serial monitor di Arduino IDE.



Gambar 3. Simulasi Keypad pada Maple Mini

Selain pada serial monitor, karakter yang ditekan pada keypad dapat ditampilkan pada *seven segment*. *Seven segment* memiliki 10 pin, 2 pin Ground dan 8 lainnya merupakan pin LED pada *seven segment* yang ditandai dengan huruf dari a hingga h. Dengan mengubah rangkaian dan program Arduino IDE dapat dituliskan⁴:



Gambar 4. Rangkaian Keypad dan Seven Segment pada Maple Mini

```

byte h=0, v=0;

const unsigned long period=50;
unsigned long kdelay=0;
const byte rows=4;
const byte columns=4;

const byte Output[rows] = {16,17,18,19};
const byte Input[columns] = {9,8,6,5};
const byte sevenseg[8]={10,7,27,26,25,11,1,4};

byte keypad()
{
    static bool no_press_flag=0;
    for(byte x=0;x<columns;x++)
    {
        if (digitalRead(Input[x])==HIGH);
        else
            break;
        if(x==(columns-1))
        {
            no_press_flag=1;
            h=0;
            v=0;
        }
    }
    if(no_press_flag==1)
    {
        for(byte r=0;r<rows;r++)
            digitalWrite(Output[r],LOW);
        for(h=0;h<columns;h++)
        {
            if(digitalRead(Input[h])==HIGH)
                continue;
            else
            {
                for (v=0;v<rows;v++)
                {
                    digitalWrite(Output[v],HIGH);
                    if(digitalRead(Input[h])==HIGH)
                    {
                        no_press_flag=0;
                        for(byte w=0;w<rows;w++)
                            digitalWrite(Output[w],LOW);
                        return v*4+h;
                    }
                }
            }
        }
    }
    return 50;
}

```

```

void setup()
{
    for(byte i=0;i<rows;i++)
    {
        pinMode(Output[i],OUTPUT);
    }

    for(byte j=0;j<=7;j++)
    {
        pinMode(sevenseg[j],OUTPUT);
    }

    for(byte s=0;s<columns;s++)
    {
        pinMode(Input[s],INPUT_PULLUP);
    }
    Serial.begin(9600);
}

void clrall()
{
    for(int i=0;i<=7;i++)
        digitalWrite(sevenseg[i],LOW);
}

void ledon()
{
    for(int i=0;i<=7;i++)
        digitalWrite(sevenseg[i],HIGH);
}

void loop()
{
    if(millis()-kdelay>period)
    {
        kdelay=millis();
        clrall();
        switch (keypad())
        {
            case 0:
            {
                Serial.println(1);
                digitalWrite(sevenseg[1],HIGH);
                digitalWrite(sevenseg[2],HIGH);
                delay(1000);
                break;
            }
            case 1:
            {
                Serial.println(2);
                ledon();
            }
        }
    }
}

```

```

        digitalWrite(sevensseg[2],LOW);
        digitalWrite(sevensseg[5],LOW);
        delay(1000);
        break;
    }
    case 2:
    {
        Serial.println(3);
        ledon();
        digitalWrite(sevensseg[4],LOW);
        digitalWrite(sevensseg[5],LOW);
        delay(1000);
        break;
    }
    case 4:
    {
        Serial.println(4);
        ledon();
        digitalWrite(sevensseg[0],LOW);
        digitalWrite(sevensseg[3],LOW);
        digitalWrite(sevensseg[4],LOW);
        delay(1000);
        break;
    }
    case 5:
    {
        Serial.println(5);
        ledon();
        digitalWrite(sevensseg[1],LOW);
        digitalWrite(sevensseg[4],LOW);
        delay(1000);
        break;
    }
    case 6:
    {
        Serial.println(6);
        ledon();
        digitalWrite(sevensseg[1],LOW);
        delay(1000);
        break;
    }
    case 8:
    {
        Serial.println(7);
        digitalWrite(sevensseg[0],HIGH);
        digitalWrite(sevensseg[1],HIGH);
        digitalWrite(sevensseg[2],HIGH);
        delay(1000);
        break;
    }
    case 9:
    {
        Serial.println(8);
        ledon();
        delay(1000);
    }

```

```

        break;
    }
    case 10:
    {
        Serial.println(9);
        ledon();
        digitalWrite(sevenseg[4],LOW);
        delay(1000);
        break;
    }
    case 3:
    {
        Serial.println("A");
        ledon();
        digitalWrite(sevenseg[3],LOW);
        delay(1000);
        break;
    }
    case 7:
    {
        Serial.println("B");
        for(int i=2;i<=6;i++)
        {
            digitalWrite(sevenseg[i],HIGH);
        }
        delay(1000);
        break;
    }

    case 11:
    {
        Serial.println("C");
        ledon();
        digitalWrite(sevenseg[1],LOW);
        digitalWrite(sevenseg[2],LOW);
        digitalWrite(sevenseg[6],LOW);
        digitalWrite(sevenseg[7],LOW);
        delay(1000);
        break;
    }

    case 13:
    {
        Serial.println(0);
        ledon();
        digitalWrite(sevenseg[6],LOW);
        delay(1000);

        break;
    }

    case 15:
    {
        Serial.println("d");
        for(int i=1;i<=4;i++)

```

```

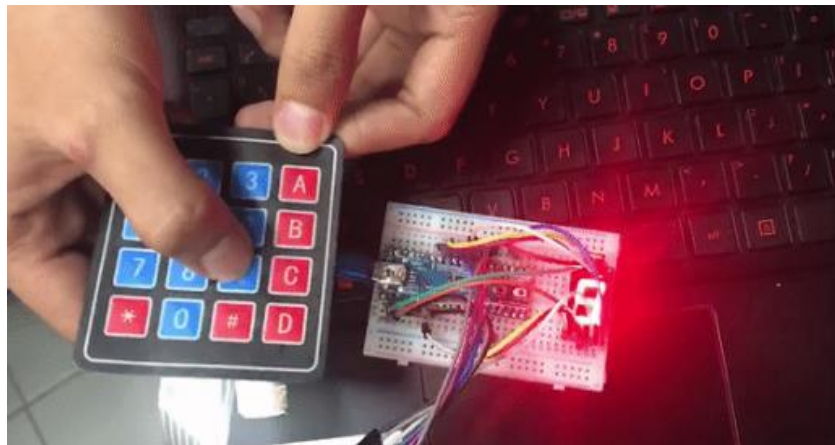
        {
            digitalWrite(sevenseg[i],HIGH);
        }
        digitalWrite(sevenseg[6],HIGH);
        delay(1000);
        break;
    }

    default:
    {
        digitalWrite(sevenseg[6],HIGH);
    }
}

}
}

```

Dengan rangkaian dan program tersebut dapat dilihat hasilnya:



Gambar 5. Simulasi Keypad dan Seven Segment pada Maple Mini

³Selengkapnya: <https://github.com/fareadh/sbm2k19-sub2-genap/blob/master/keypad/keypad.ino>.

⁴Selengkapnya: https://github.com/fareadh/sbm2k19-sub2-genap/blob/master/keypad_7segment/keypad_7segment/keypad_7segment.ino.

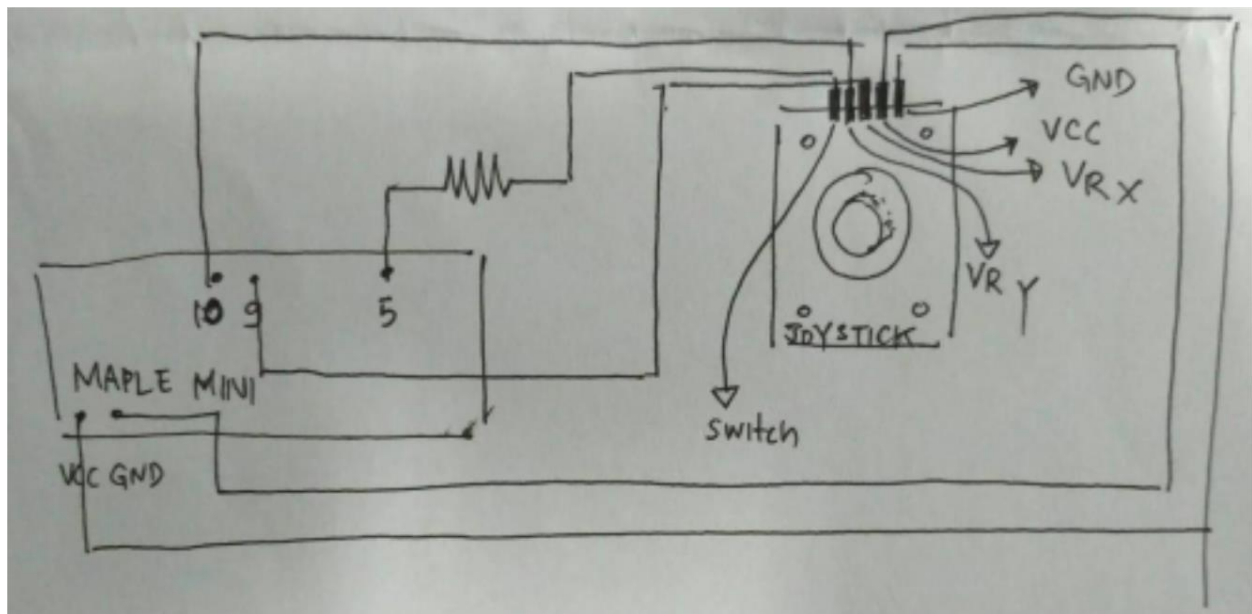
Bab 4

Joystick

Joystick adalah perangkat input yang terdiri dari tongkat yang berputar di suatu bidang alas dan memberikan keluaran sudut atau arahnya ke perangkat yang dikendalikannya. Joystick, juga dikenal sebagai *control column*, adalah perangkat kontrol utama di kokpit pesawat sipil dan militer. Seringkali memiliki saklar/switch tambahan untuk mengontrol berbagai aspek penerbangan pesawat. Joystick elektronik dua sumbu ditemukan oleh C. B. Mirick di Laboratorium Penelitian Angkatan Laut Amerika Serikat (*United States Naval Research Laboratory*) dan dipatenkan pada tahun 1926 (U.S. Patent no. 1,597,416).

4.1. Rangkaian

Joystick yang digunakan pada laporan ini memiliki 5 pin yaitu V_{CC} , Ground, V_{RX} , V_{RY} dan Switch. V_{RX} merupakan isyarat analog masukan yang menunjukkan posisi sumbu x pada joystick, V_{RY} merupakan isyarat analog masukan yang menunjukkan posisi sumbu y pada joystick dan Switch berupa isyarat digital masukan yang menunjukkan kondisi ON/OFF saklar pada joystick. Pin yang digunakan pada Maple Mini yaitu untuk V_{RX} pada pin 9, V_{RY} pada pin 10 dan Switch pada pin 5.



Gambar 6. Rangkaian Joystick pada Maple Mini

4.2. Program

Untuk pin Switch harus diberi input *pull-up* karena Switch tersebut menghubungkan pin yang digunakan untuk Switch dengan Ground. Hal tersebut akan mengakibatkan ketika Switch ditekan akan membaca logika LOW. Jika tidak diberi input *pull-up* yang akan terjadi adalah nilai yang tidak menentu pada bacaan pin Switch pada keadaan *idle*. Pada Arduino IDE dapat dituliskan⁵:

```
#define THRESHOLD 100

double dt_last = 0.0;
String SWITCH = "OFF";

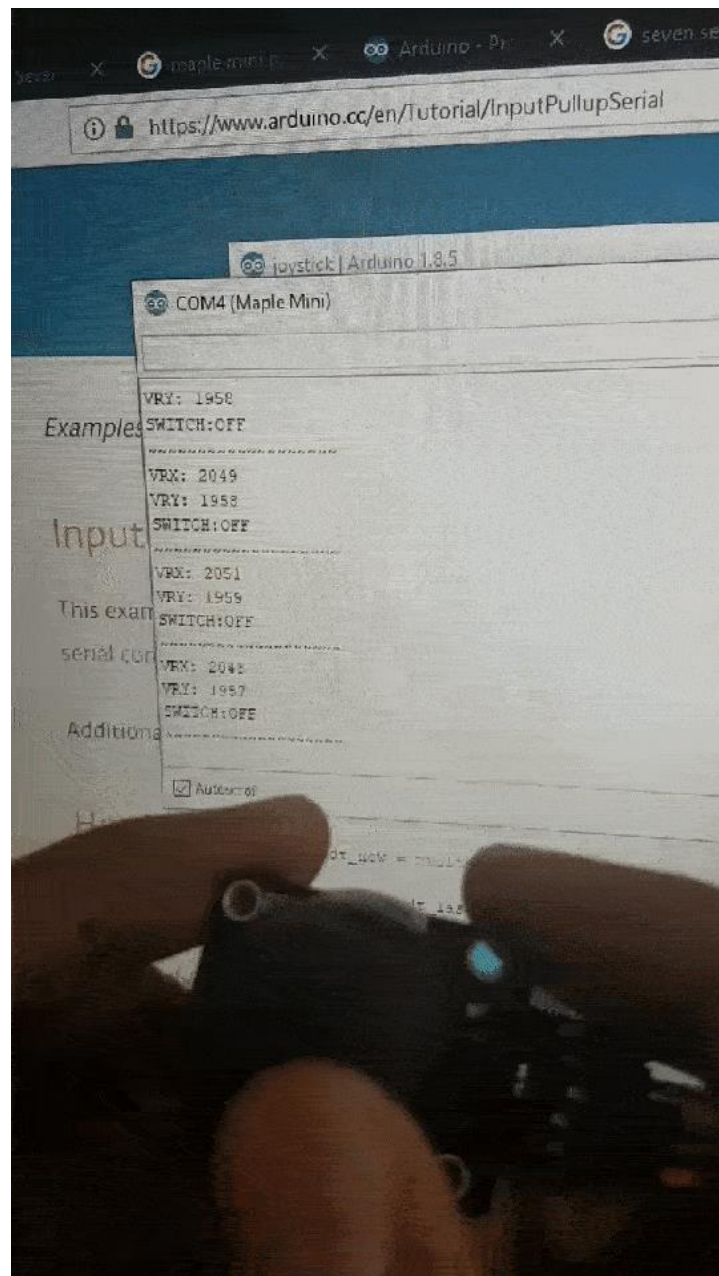
void setup() {
  pinMode(5, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop() {
  auto vry = analogRead(9);
  auto vrx = analogRead(10);
  if(!digitalRead(5)){
    SWITCH = "ON";
  }
  else{
    SWITCH = "OFF";
  }
  auto dt_now = millis();
  if((dt_now - dt_last) > THRESHOLD){
    Serial.print("VRX: ");
    Serial.println(vrx);
    Serial.print("VRY: ");
    Serial.println(vry);
    Serial.print("SWITCH:");
    Serial.println(SWITCH);
    Serial.println("~~~~~");
    dt_last = dt_now;
  }
}
```

⁵Selengkapnya: <https://github.com/fareadh/sbm2k19-sub2-genap/blob/master/joystick/joystick.ino>.

4.3. Simulasi

Kemudian program dan rangkaian tersebut direalisasikan pada Maple Mini. Resolusi bacaan ADC (Analog to Digital Converter) pada Maple Mini sebesar 12 bit sehingga nilai bacaan ADC akan membentang dari 0 hingga 4095. Nilai V_{RX} dan V_{RY} pada joystick keadaan *idle* kira-kira sebesar 2048. Jika joystick digerakan ke sumbu x positif maka V_{RX} akan bernilai diantara 2048 hingga 4095. Jika joystick digerakan ke sumbu x negatif maka V_{RX} akan bernilai diantara 0 hingga 2048. Jika joystick digerakan ke sumbu y positif maka V_{RY} akan bernilai diantara 2048 hingga 4095. Jika joystick digerakan ke sumbu y negatif maka V_{RY} akan bernilai diantara 0 hingga 2048. Jika Switch ditekan akan bernilai LOW sebaliknya jika Switch dalam keadaan *idle* akan bernilai HIGH.



Gambar 7. Simulasi Joystick pada Maple Mini

Bab 5

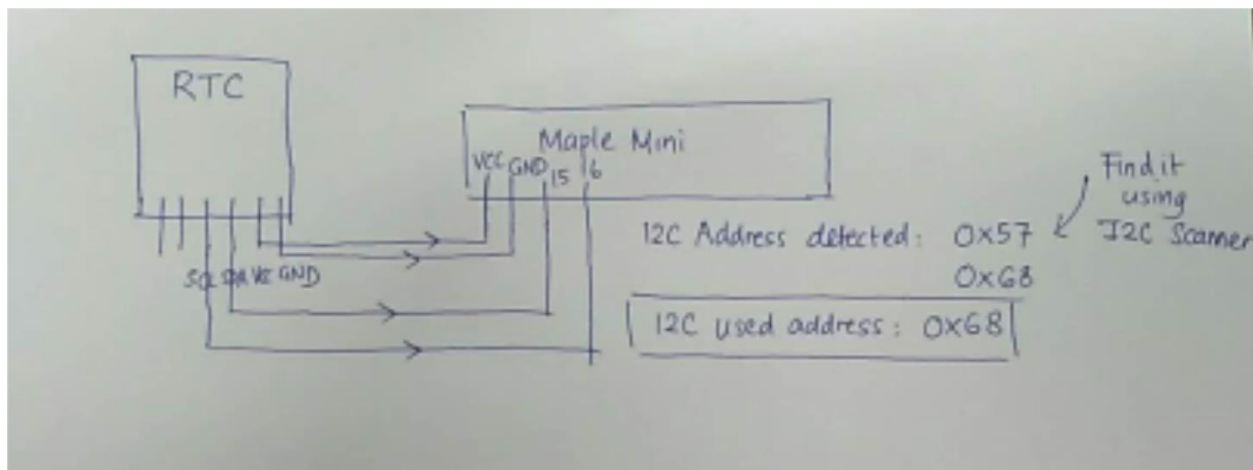
Realtime Clock

"Time is an illusion."
– Albert Einstein

Waktu merupakan hal yang esensial dalam kehidupan umat manusia. Dalam pengukuran dan perhitungan seperti kecepatan, tingkat kebusukan bahan makanan dan reaksi kimia menggunakan waktu untuk menentukan hasilnya. Keakuratan perhitungan waktu dapat membantu perhitungan sehingga kesalahan yang didapatkan menjadi kecil. Dalam dunia mikrokontroler terdapat modul yang digunakan untuk menghitung waktu secara akurat yaitu Realtime Clock.

5.1. Rangkaian

Modul Realtime Clock yang digunakan pada laporan ini menggunakan IC DS3231. IC DS3231 adalah Real Time Clock berbiaya rendah dan sangat akurat yang dapat mempertahankan jam, menit, dan detik, serta informasi hari, bulan, dan tahun. Juga, ia memiliki kompensasi otomatis untuk tahun kabisat dan berbulan-bulan dengan kurang dari 31 hari. Modul Realtime Clock ini dapat bekerja pada 3.3 atau 5 V. Terdapat input eksternal baterai berupa CR2032 3V yang dapat memberi daya pada modul dan mempertahankan informasi selama lebih dari setahun. Modul ini menggunakan protokol komunikasi I²C sehingga memiliki pin V_{CC}, Ground, SDA dan SCL. Pada Maple Mini SDA terletak pada pin 15 dan SCL terletak pada pin 16.



Gambar 8. Rangkaian Realtime Clock pada Maple Mini

5.2. Program

Sebelum merancang program Realtime Clock (RTC) pada Maple Mini dilakukan pencarian alamat I²C dengan I²C Scanner sehingga akan didapatkan alamat I²C yang akan digunakan. Program yang akan dirancang pada Maple Mini akan membuat waktu akan terulang sesuai parameter pada program. Pada Arduino IDE dapat dituliskan⁶:

```
#include "Wire.h"
#define DS3231_I2C_ADDRESS 0x68

byte decToBcd(byte val)
{
  return( (val/10*16) + (val%10) );
}

byte bcdToDec(byte val)
{
  return( (val/16*10) + (val%16) );
}

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  setDS3231time(30,31,11,2,25,3,19);
}

void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek, byte
dayOfMonth, byte month, byte year)
{
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0);
  Wire.write(decToBcd(second));
  Wire.write(decToBcd(minute));
  Wire.write(decToBcd(hour));
  Wire.write(decToBcd(dayOfWeek));
  Wire.write(decToBcd(dayOfMonth));
  Wire.write(decToBcd(month));
  Wire.write(decToBcd(year));
  Wire.endTransmission();
}

void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0);
  Wire.endTransmission();
  Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
  *second = bcdToDec(Wire.read() & 0x7f);
```

```

*minute = bcdToDec(Wire.read());
*hour = bcdToDec(Wire.read() & 0x3f);
*dayOfWeek = bcdToDec(Wire.read());
*dayOfMonth = bcdToDec(Wire.read());
*month = bcdToDec(Wire.read());
*year = bcdToDec(Wire.read());
}

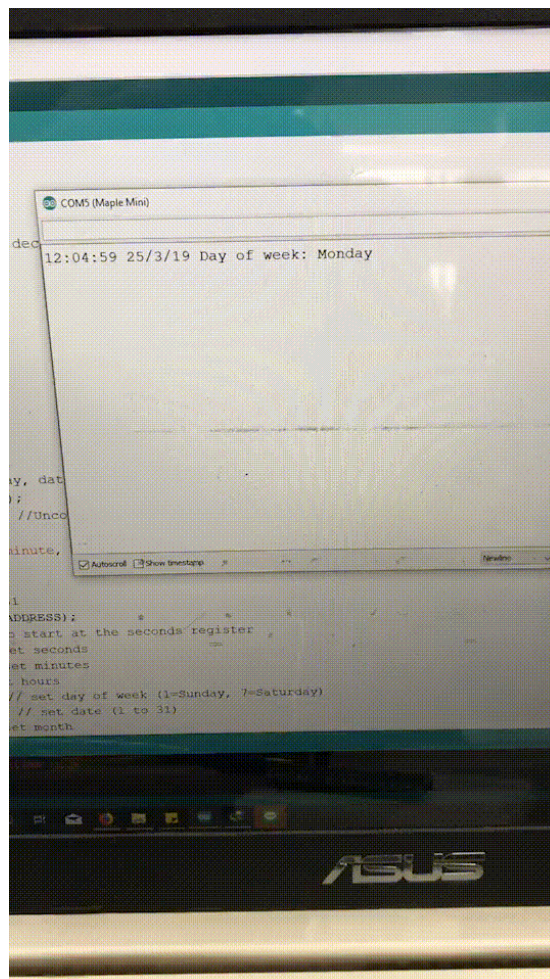
void displayTime()
{
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
&year);
Serial.print(hour, DEC);
Serial.print(":");
if (minute<10)
{
Serial.print("0");
}
Serial.print(minute, DEC);
Serial.print(":");
if (second<10)
{
Serial.print("0");
}
Serial.print(second, DEC);
Serial.print(" ");
Serial.print(dayOfMonth, DEC);
Serial.print("/");
Serial.print(month, DEC);
Serial.print("/");
Serial.print(year, DEC);
Serial.print(" Day of week: ");
switch(dayOfWeek){
case 1:
Serial.println("Sunday");
break;
case 2:
Serial.println("Monday");
break;
case 3:
Serial.println("Tuesday");
break;
case 4:
Serial.println("Wednesday");
break;
case 5:
Serial.println("Thursday");
break;
case 6:
Serial.println("Friday");
break;
case 7:
Serial.println("Saturday");
break;
}
}

```

```
}  
}  
  
void loop()  
{  
  displayTime();  
  delay(1000);  
}
```

5.3. Simulasi

Program dan Rangkaian yang dirancang diimplementasikan pada Maple Mini. Waktu yang dihitung oleh Realtime Clock ditampilkan pada serial monitor. Hasil simulasi akan didapatkan seperti gambar di bawah ini.



Gambar 9. Simulasi Realtime Clock pada Maple Mini

⁵Selengkapnya: <https://github.com/fareadh/sbm2k19-sub2-genap/blob/master/rtc/rtc.ino>.

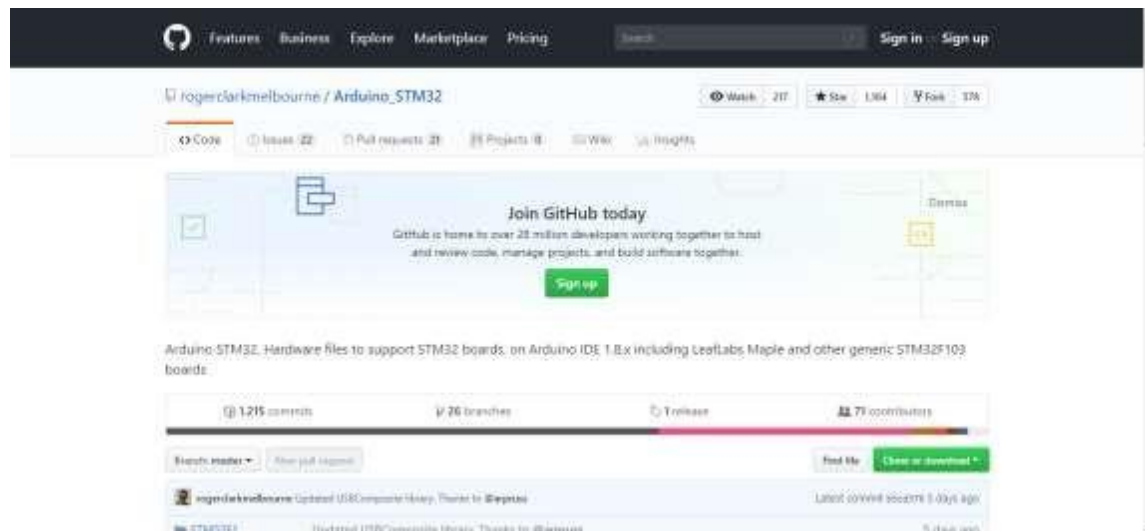
Daftar Pustaka

- [1] Dejan. *Arduino and DS3231 Real Time Clock Tutorial*. How to Mechatronics. Available at: <https://howtomechatronics.com/tutorials/arduino/arduino-ds3231-real-time-clock-tutorial/> [Accessed 29 Mar. 2019].
- [2] Zeller Jr., Tom. *A Great Idea That's All in the Wrist*. New York Times. Available at: <https://www.nytimes.com/2005/06/05/weekinreview/a-great-idea-thats-all-in-the-wrist.html> [Accessed 29 Mar. 2019].
- [3] Pradana, Sunu. *Uji Maple Mini Compatible (STM32F103C8T6)*. Sunu Pradana. Available at: <https://sunupradana.info/tkr/2016/08/16/uji-maple-mini-compatible-stm32f103c8t6/> [Accessed 29 Mar. 2019].
- [4] Nakamura, Shuji, Stephen Pearton, and Gerhard Fasol. *The blue laser diode: the complete story*. Springer Science & Business Media, 2000.
- [5] Star-Ledger Staff. *Monmouth Man, Inventor of Touch-tone Keypad, Dies at 94*. NJ.com. Available at: https://www.nj.com/news/2013/02/monmouth_man_inventor_of_touch.html [Accessed 29 Mar. 2019].

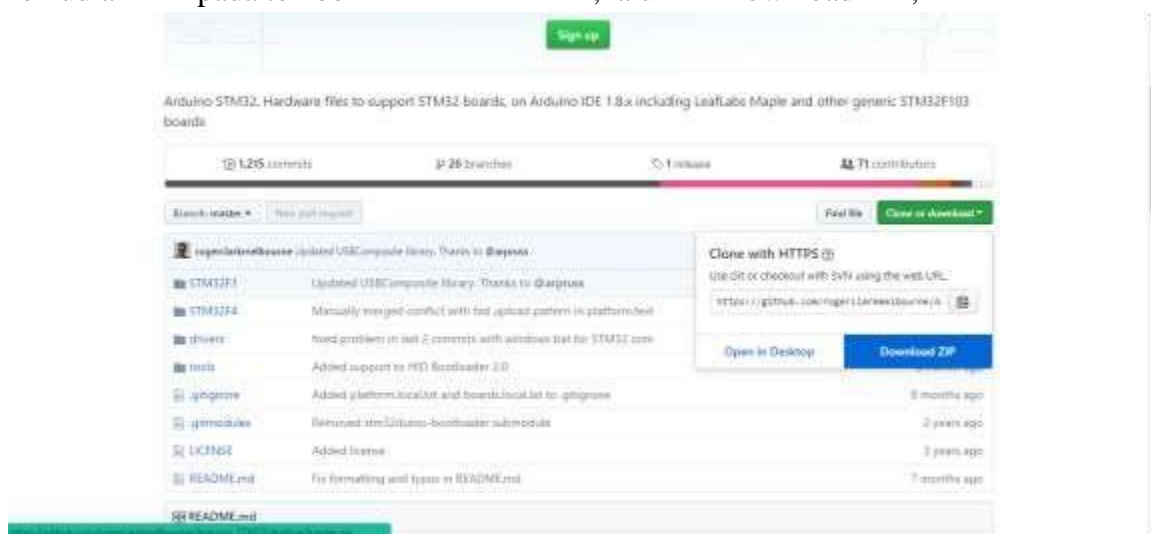
Appendix A

Setup Maple Mini pada Arduino

Arduino IDE merupakan *software open-source*, sehingga orang-orang dapat menyumbangkan berbagai macam tambahan ke dalamnya agar fungsinya semakin lengkap dan dapat digunakan dengan perangkat tambahan selain Arduino. Dikarenakan kita nantinya akan menggunakan perangkat yang didasarkan pada *board* STM32F103, maka kita memerlukan *library-library* tambahan untuk menyesuaikannya. Pertama, kita perlu mengunduh beberapa *library* tambahan ini dari internet. Buka peramban web anda, lalu buka tautan ini: https://github.com/rogerclarkmelbourne/Arduino_STM32. Akan muncul laman seperti berikut



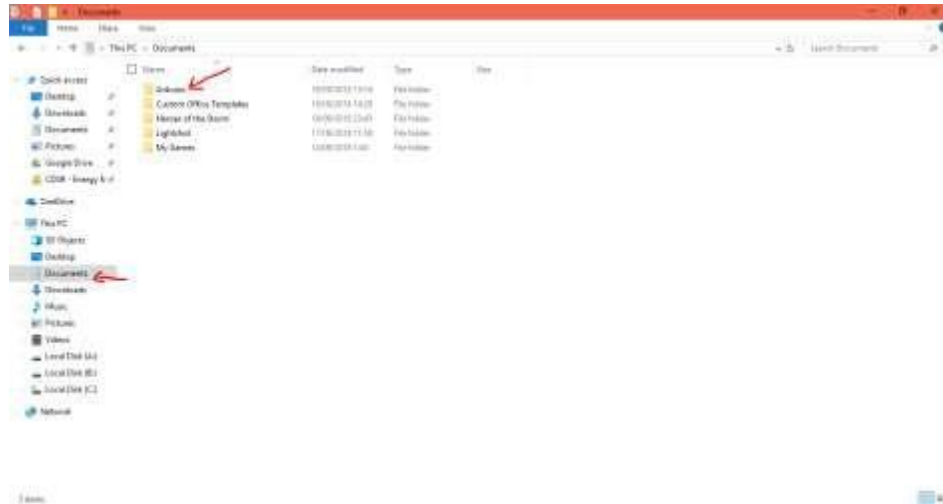
Kemudian klik pada tombol **Clone or download**, lalu klik Download ZIP,



Peramban akan mengunduh *file* bernama Arduino_STM32-master.zip.

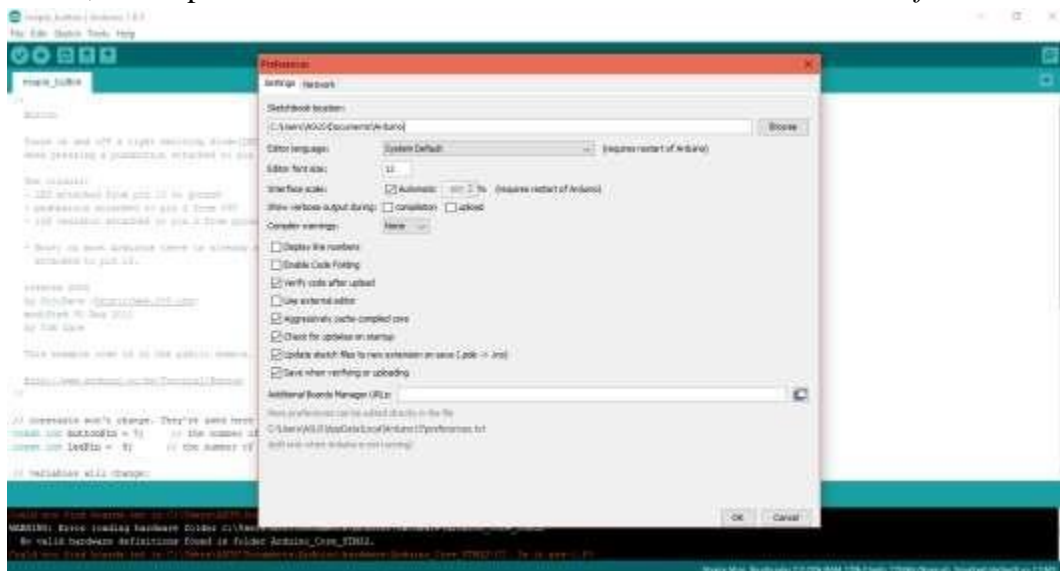
Setelah mengunduh, ekstrak *file* ZIP, yang akan mengeluarkan folder bernama Arduino_STM32- master.

Salin folder tersebut, lalu buka folder Arduino di Documents,



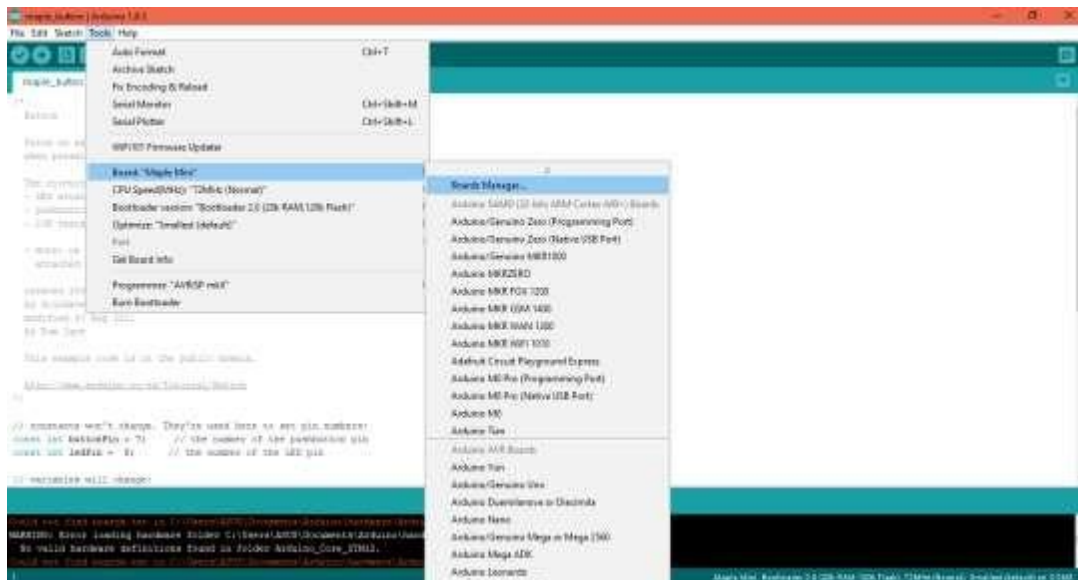
Tempel folder tersebut di dalam folder Documents/Arduino/hardware. Jika belum ada, buat folder bernama *hardware* kemudian tempel kedua folder yang telah disalin ke dalam folder *hardware*.

Setelah itu, buka aplikasi Arduino IDE. Buka menu *File* lalu klik menu *Preferences*.

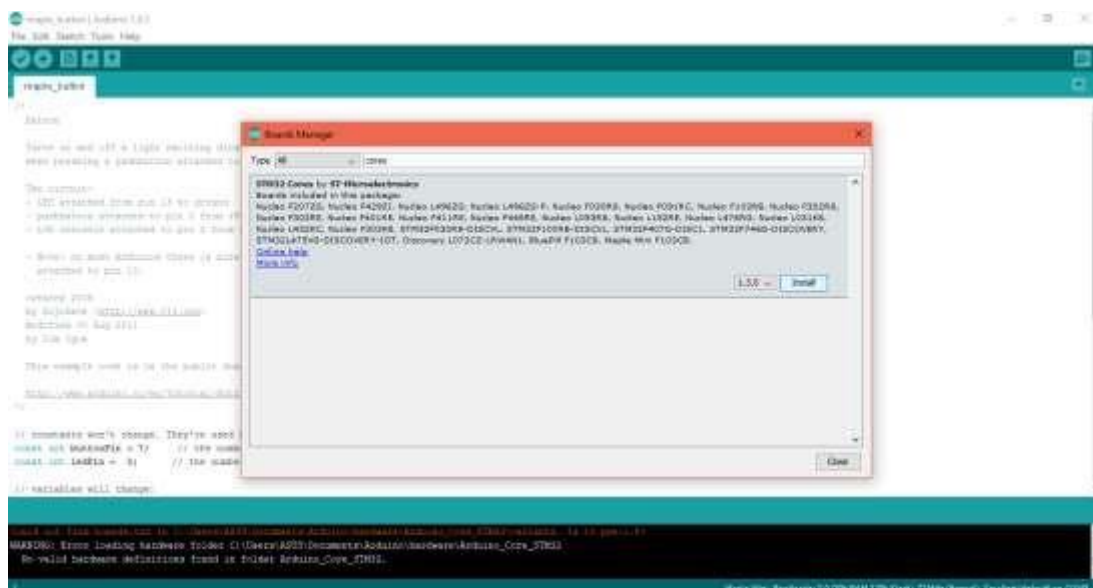


Pada baris *Additional Boards Manager URLs*, isi dengan tautan berikut ini, https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json. Lalu klik ok.

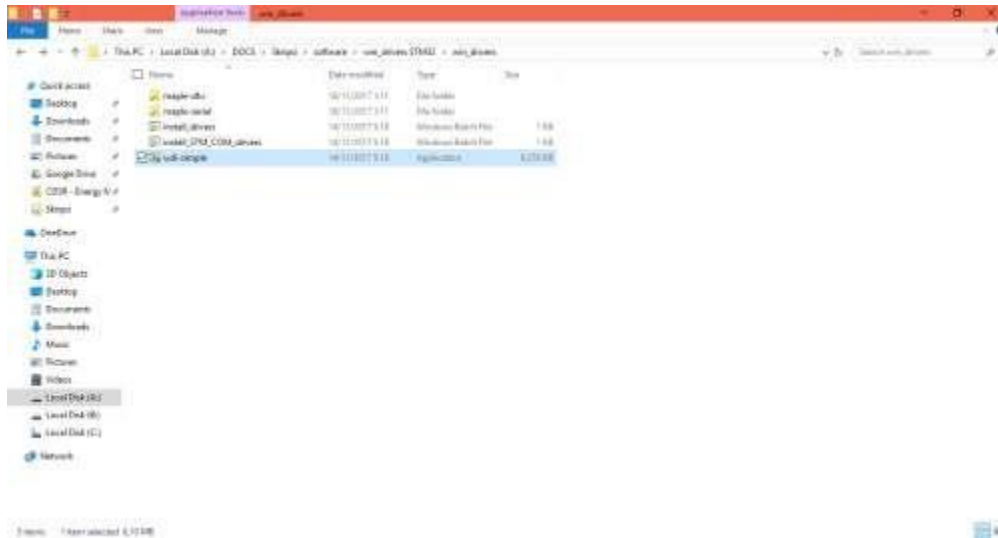
Kemudian buka menu *Tools*, klik bagian *Board*, lalu klik *Boards Manager*.



Akan muncul jendela *Boards Manager*, pada filter ketik STM32 Cores, kemudian setelah muncul STM32 Cores klik *install*. Tunggu hingga selesai.



Agar komputer kita dapat mengenali perangkat *board* Maple Mini pada saat dicolokkan USB, maka diperlukan sebuah *driver* untuk mengatur antarmuka antara komputer dan *board* Maple. Driver dapat diunduh pada tautan: <http://ugm.id/DriverMaple> (harus *login* dengan email UGM). Setelah selesai mengunduh, ekstrak *file* “win_drivers STM32.zip”.



Jalankan program *wdi-simple.exe* untuk memasang *driver*. Akan muncul sebuah jendela *command prompt* seperti berikut,



Selama proses pemasangan berlangsung akan muncul pop-up pemasangan *driver* seperti pada tutorial 1 saat memasang Arduino IDE, klik *install* pada semua pop-up. Jendela Pop-up akan muncul mirip seperti gambar berikut namun tentunya dengan keterangan yang berbeda.

