LAPORAN AKHIR

FINAL PROJECT SISTEM TERTANAM

Disusun oleh:

Iqbal Muchlis 5024201073

Dosen Pembimbing:

Eko Pramunanto, S.T., M.T.

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS

TEKNIK KOMPUTER

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2023

I. PENJELASAN TUGAS

Rancanglah sistem untuk menampilkan jam, kalender, suhu dan alarm dengan karakter pada dot matrix LED 8 x 32 (4 buah matrix LED 8x8), dengan input keyboard USB. Suhu diukur dengan menggunakan sensor suhu analog (seperti LM35 atau yang sejenisnya).

Spesifikasi fitur:

a. mode : run, set jam, set tanggal, set alarm.

b. select set : jam, menit, detik / tgl, bln, thn,

c. 3 waktu alarm dengan text (wajib) + buzzer (optional nilai plus):

- alarm 1: Display NRP

- alarm 2: Display NRP + Nama

- alarm 3: Display text yang diinputkan

Waktu aktif untuk alarm 1, 2 dan 3 bisa diset dengan tanggal, jam, menit dan durasinya dalam detik.

d. Tampilan:

- Kecerahan diatur sesuai kecerahan lingkungan dengan sensor cahaya.

- Jam, menit, detik

- Pada setiap detik ke 10 dan 40 tampilkan tgl-bln-thn selama 3 detik

- Pada setiap detik ke 13 dan 43 tampilkan suhu dengan keterangan ^{o}C (derajad Celcius)

- Pada saat waktu tepat sama dengan waktu seting alarm tampilkan text alarm sesuai durasi setingnya.

- Kecepatan geser tampilan text panjang (running text) adalah 0.5 detik per kolom matrix LED.

- Sebagai tambahan nilai : tampilan berkedip saat berada pada mode seting (input dari keyboard).

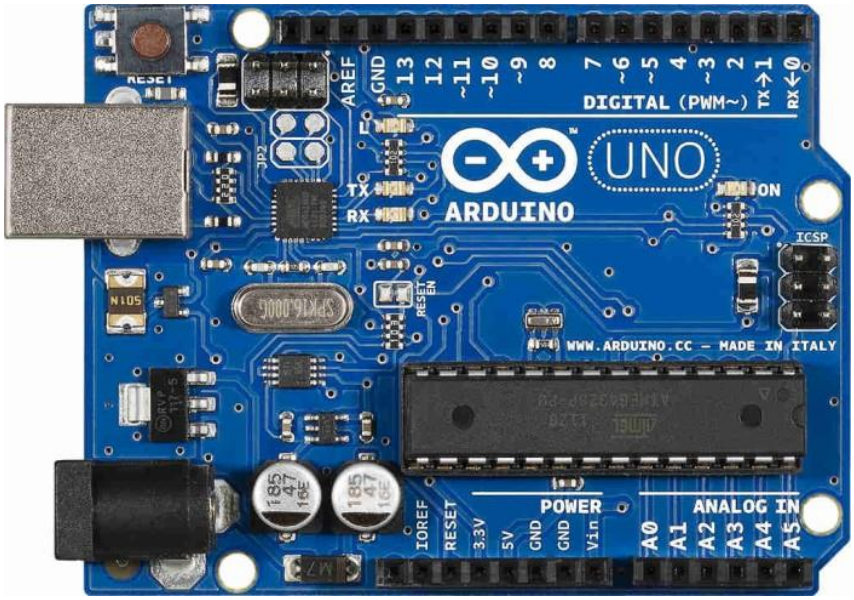
Output:

- 8x32 dot matrix LED

II. ALAT DAN BAHAN

Alat dan bahan yang saya pakai untuk final project ini adalah:

- Arduino UNO

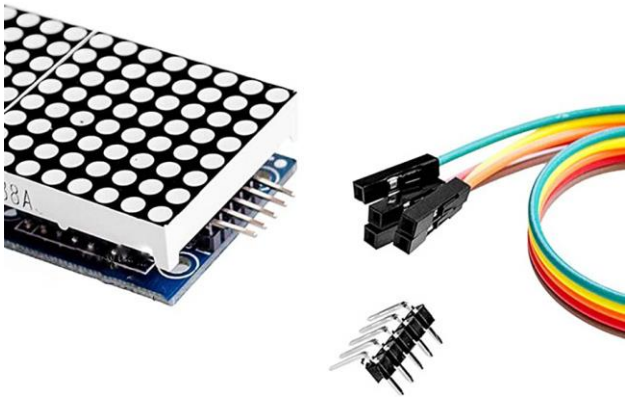


- Keyboard PS/2 Protocol

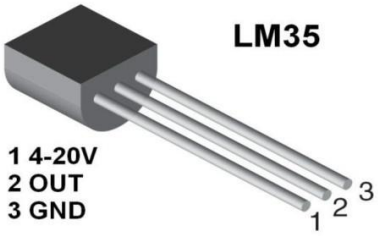
- Female PS/2 Socket

- Active Buzzer

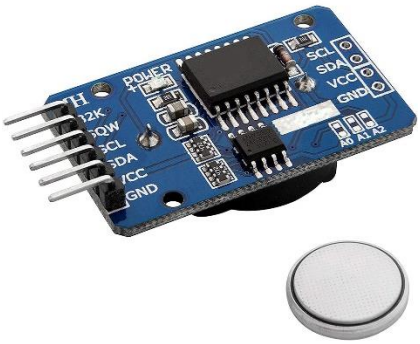
- MAX7219 LED Dot Matrix Module 4-IN-1 32x8



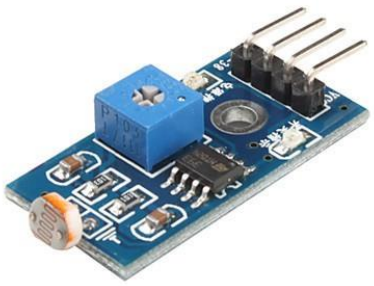
- LM35



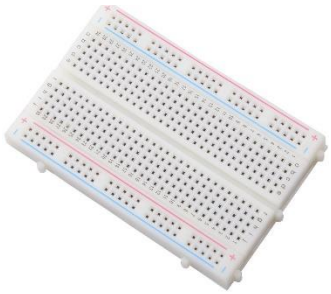
- Real time clock (DS3232)



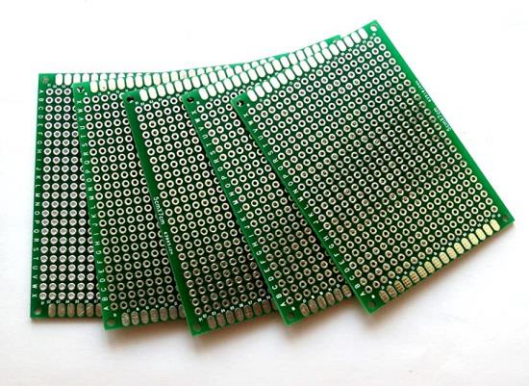
- LDR (Light Sensor Module)



- Breadboard (Untuk bereksperimen)



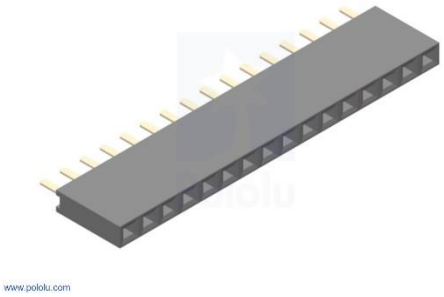
- PCB



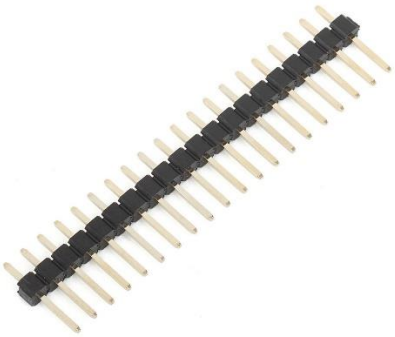
- Solder dan timahnya

- Kaki PCB

- Female to Male Pin Header



- Male to Male Pin Header



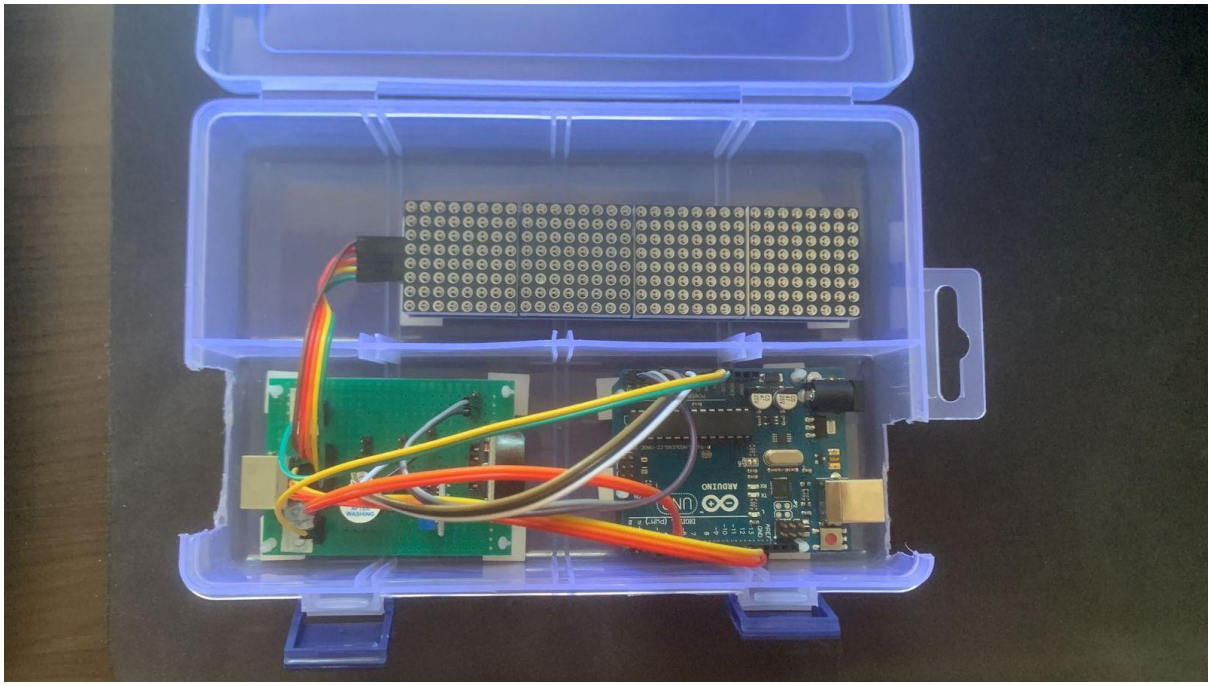
- Kabel
  - Male to female

  - Female to female

- Male to male (untuk bereksperimen)

III. RANGKAIAN

Berikut merupakan rangkaian yang telah saya rancang:

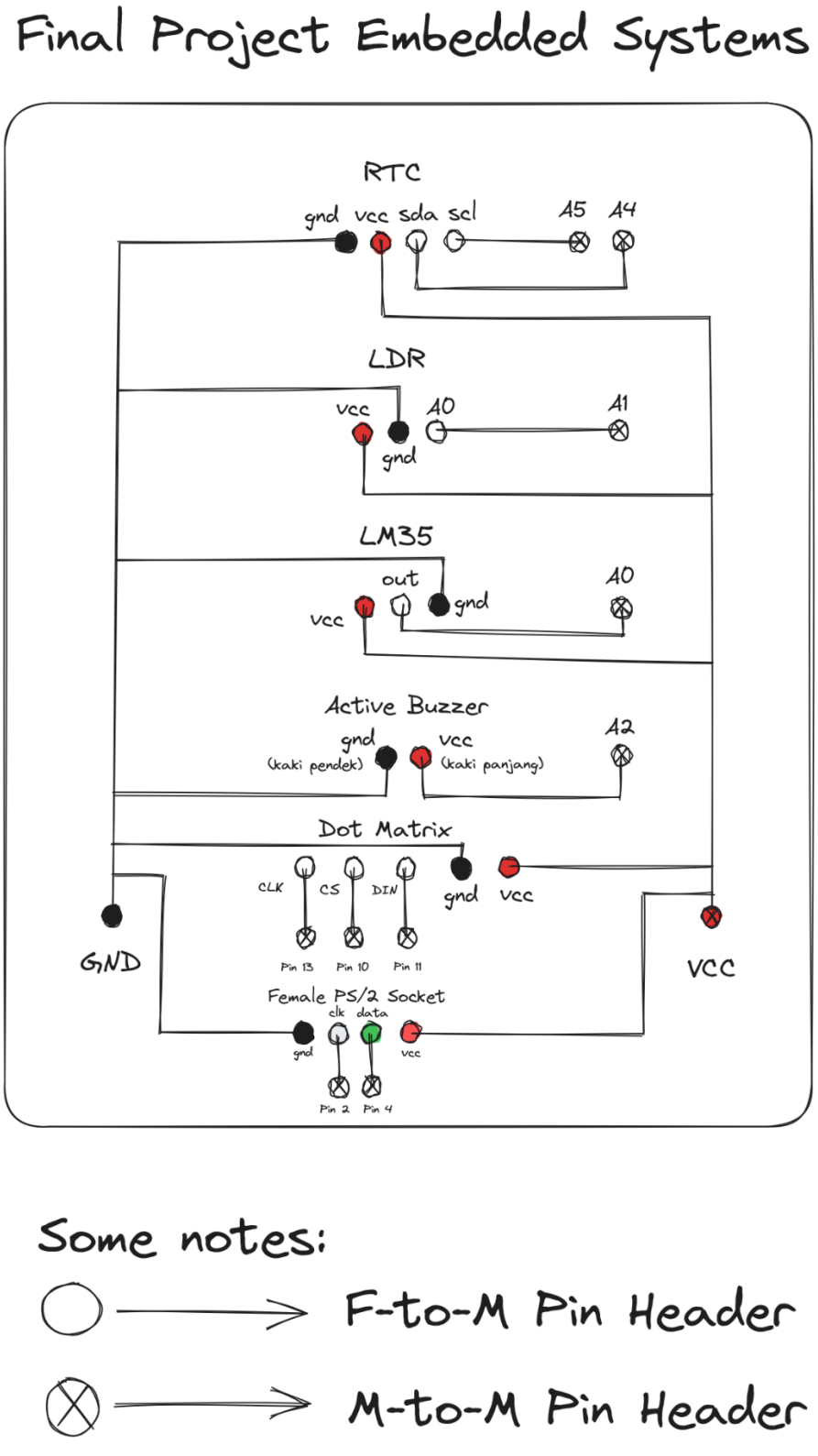


Detail rangkaian:

- Dot matrix
  - VCC disambungkan pada 5v di arduino
  - GND disambungkan pada GND di arduino
  - DIN disambungkan pada pin 11 di arduino
  - CS disambungkan pada pin 10 di arduino
  - CLK disambungkan pada pin 13 di arduino
- LDR
  - VCC disambungkan pada 5v di arduino
  - GND disambungkan pada GND di arduino
  - A0 disambungkan pada A1 di arduino
- Active Buzzer
  - Pin 1 disambungkan pada GND di arduino
  - Pin 2 disambungkan pada 8 di arduino
- RTC
  - VCC disambungkan pada 5v di arduino
  - GND disambungkan pada GND di arduino

    - SDA disambungkan pada A4 di arduino
    - SCL disambungkan pada A5 di arduino
- LM35
    - 4-20V disambungkan pada 5v di arduino
    - GND disambungkan pada GND di arduino
    - OUT disambungkan pada A0 di arduino
- Female PS/2 Socket
    - VCC disambungkan pada VCC di arduino
    - GND disambungkan pada GND di arduino
    - CLK disambungkan pada Pin 2 di arduino
    - DATA disambungkan pada Pin 4 di arduino

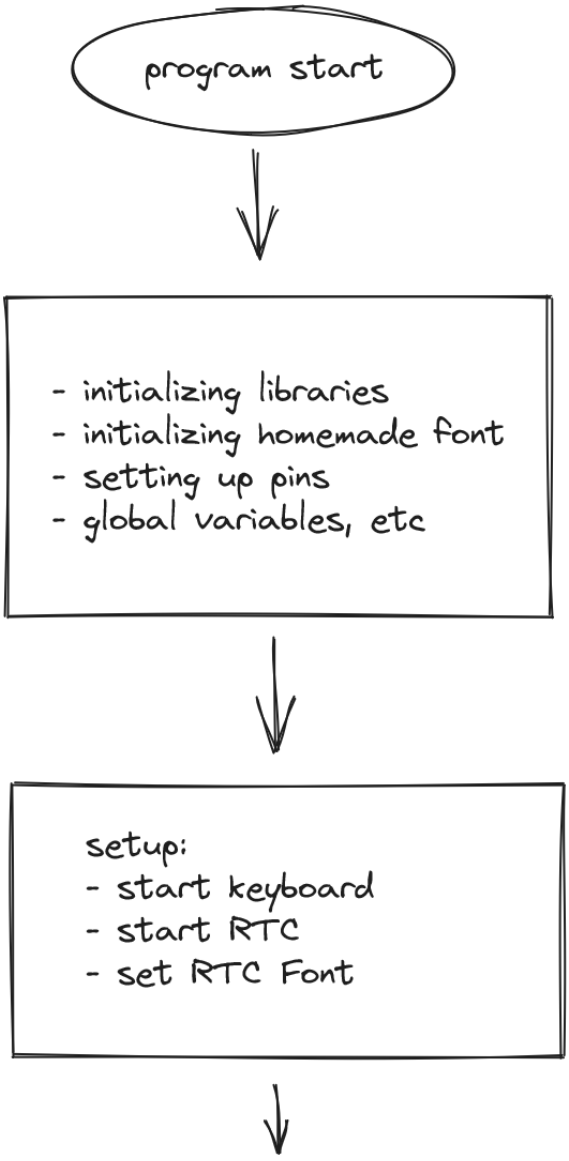
Rancangan pada PCB:



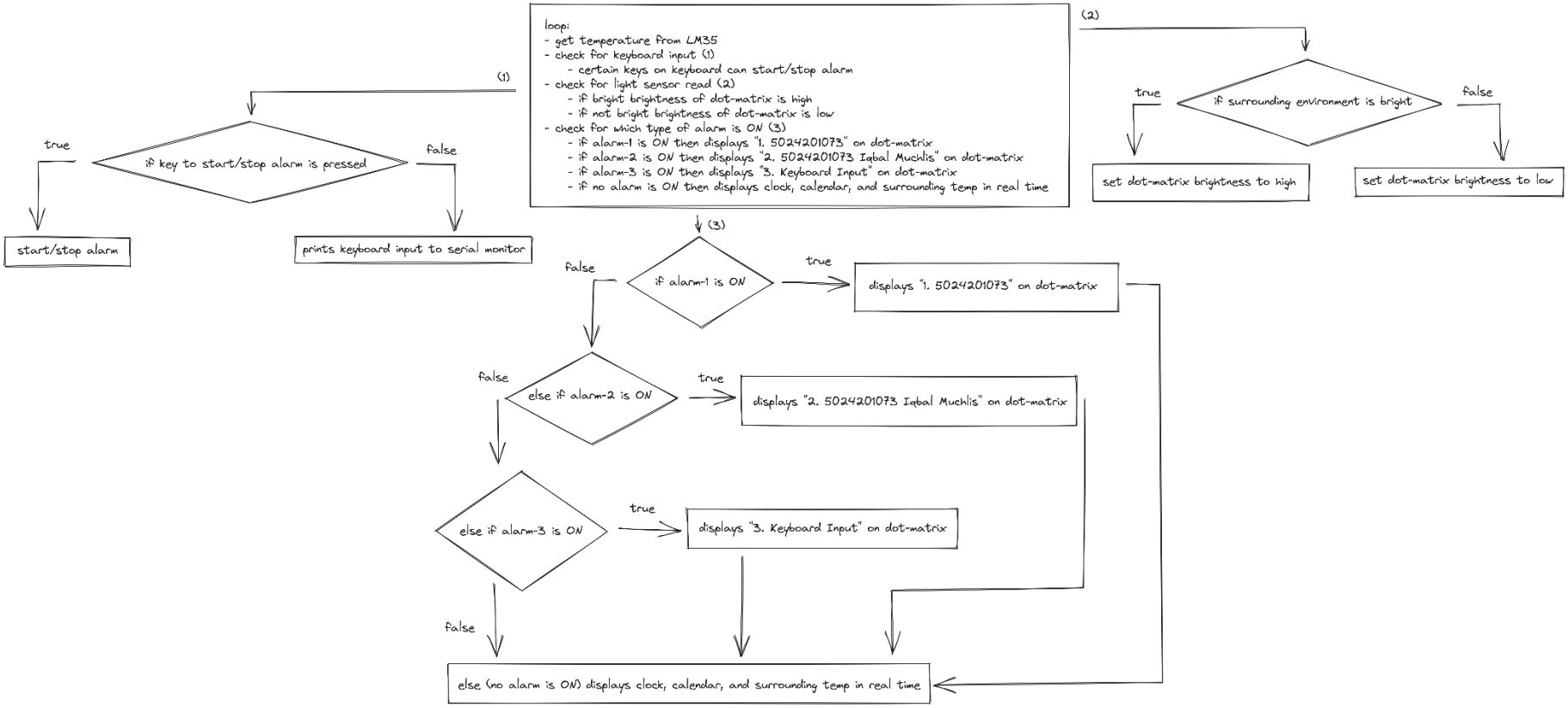
IV. HASIL

- Diagram blok:

  - Part 1:



  - Part 2:


loop:
- get temperature from LM35
- check for keyboard input (1)
  - certain keys on keyboard can start/stop alarm
- check for light sensor read (2)
  - if bright brightness of dot-matrix is high
  - if not bright brightness of dot-matrix is low
- check for which type of alarm is ON (3)
  - if alarm-1 is ON then displays "1. 5024201073" on dot-matrix
  - if alarm-2 is ON then displays "2. 5024201073 Iqbal Muchlis" on dot-matrix
  - if alarm-3 is ON then displays "3. Keyboard Input" on dot-matrix
  - if no alarm is ON then displays clock, calendar, and surrounding temp in real time
(1)
if key to start/stop alarm is pressed
true
false
start/stop alarm
prints keyboard input to serial monitor
(2)
if surrounding environment is bright
true
false
set dot-matrix brightness to high
set dot-matrix brightness to low
(3)
if alarm-1 is ON
true
false
displays "1. 5024201073" on dot-matrix
else if alarm-2 is ON
true
false
displays "2. 5024201073 Iqbal Muchlis" on dot-matrix
else if alarm-3 is ON
true
false
displays "3. Keyboard Input" on dot-matrix
else (no alarm is ON) displays clock, calendar, and surrounding temp in real time

- Dengan menggunakan library:
    - MD_MAX72XX (https://github.com/MajicDesigns/MD_MAX72XX)
        - untuk mengatur display dot matrix
    - MD_Parola (https://github.com/MajicDesigns/MD_Parola)
        - untuk mengatur, memunculkan tampilan, scrolling text, dll pada display dot matrix
    - Streaming (https://github.com/janelia-arduino/Streaming)
        - sehingga dapat melakukan cout << variable << endl seperti pada c++
        - dilakukan dengan cara Serial << variable << endl
    - TimeLib (https://github.com/PaulStoffregen/Time)
        - untuk mendapatkan time
    - DS3232RTC (https://github.com/JChristensen/DS3232RTC)
        - untuk mengatur real time clock
    - LM35 (https://github.com/wilmouths/LM35)
        - untuk mengatur sensor temperature agar lebih mudah
    - PS2Keyboard.h (https://github.com/PaulStoffregen/PS2Keyboard)
        - untuk memproses input keyboard
- Hasil:
    - dapat melakukan semua yang diminta dari detail tugas kecuali:
      - setting alarm dengan input keyboard
      - untuk alarm-3 (memasukkan input keyboard dan menampilkannya pada dot-matrix) juga belum dapat saya selesaikan

- Code:

```
// dot matrix
#include <MD_Parola.h>   //
https://github.com/MajicDesigns/MD_Parola
#include <MD_MAX72xx.h>  //
https://github.com/MajicDesigns/MD_MAX72XX)
#include <SPI.h>
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 4
#define CLK_PIN 13
#define DATA_PIN 11
#define CS_PIN 10
MD_Parola P = MD_Parola(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);
// real time clock
```

```
#include <Streaming.h>  // https://github.com/janelia-
arduino/Streaming
#include <TimeLib.h>    // https://github.com/PaulStoffregen/Time
#include <DS3232RTC.h>  // https://github.com/JChristensen/DS3232RTC
DS3232RTC myRTC;
// temp sensor
#include <LM35.h>  // https://github.com/wilmouths/LM35
LM35 lm35(A0);
// light sensor
#define LIGHT_SENSOR A1
// buzzer
#define BUZZER A2
// keyboard
#include <PS2Keyboard.h>
#define DATAPIN 4
#define IRQPIN 2
PS2Keyboard keyboard;

// for snprintf
char buffer[60];

// new font
MD_MAX72XX::fontType_t newFont[] PROGMEM = {
  0,                                  // 0
  0,                                  // 1
  0,                                  // 2
  0,                                  // 3
  0,                                  // 4
  0,                                  // 5
  0,                                  // 6
  0,                                  // 7
  0,                                  // 8
  0,                                  // 9
  0,                                  // 10
  0,                                  // 11
  0,                                  // 12
  0,                                  // 13
  0,                                  // 14
  0,                                  // 15
  0,                                  // 16
  0,                                  // 17
  0,                                  // 18
  0,                                  // 19
  0,                                  // 20
  0,                                  // 21
  0,                                  // 22
  0,                                  // 23
  0,                                  // 24
```

```
  0,                                  // 25
  0,                                  // 26
  0,                                  // 27
  0,                                  // 28
  0,                                  // 29
  0,                                  // 30
  0,                                  // 31
  1, 0,                               //
32                                    - space
  1, 92,                              // 33
  3, 12, 0, 12,                       // 34
  5, 40, 124, 40, 124, 40,            // 35
  4, 76, 146, 255, 100,               // 36
  4, 36, 16, 8, 36,                   // 37
  4, 108, 82, 42, 84,                 // 38
  1, 6,                               // 39
  2, 60, 66,                          // 40
  2, 66, 60,                          // 41
  3, 4, 14, 4,                        // 42
  5, 16, 16, 124, 16, 16,             // 43
  2, 64, 32,                          // 44
  2, 16, 16,                          // 45
  1, 32,                              // 46
  5, 64, 32, 16, 8, 4,                // 47
  4, 56, 68, 68, 56,                  // 48                    - 0
  2, 8, 124,                          // 49                          -
1
  4, 72, 100, 84, 72,                 // 50                    - 2
  3, 84, 84, 124,                     // 51                       - 3
  3, 16, 24, 124,                     // 52                        - 4
  3, 92, 84, 116,                     // 53                       - 5
  3, 124, 84, 116,                    // 54                      - 6
  3, 4, 4, 124,                       // 55                          - 7
  3, 124, 84, 124,                    // 56                      - 8
  3, 92, 84, 124,                     // 57                       - 9
  1, 36,                              //
58                                    - :
  2, 64, 36,                          // 59
  3, 16, 40, 68,                      // 60
  3, 40, 40, 40,                      // 61
  3, 68, 40, 16,                      // 62
  4, 2, 89, 9, 6,                     // 63
  7, 28, 34, 89, 85, 93, 46, 20,      // 64
  3, 120, 20, 120,                    // 65
  3, 124, 84, 40,                     // 66
  3, 56, 68, 68,                      // 67
  3, 124, 68, 56,                     // 68
  3, 124, 84, 84,                     // 69
```

```
  3, 124, 20, 20,                 // 70
  4, 56, 68, 84, 116,             // 71
  4, 124, 16, 16, 124,            // 72
  1, 124,                         // 73
  3, 64, 64, 124,                 // 74
  3, 124, 40, 68,                 // 75
  3, 124, 64, 64,                 // 76
  5, 124, 8, 16, 8, 124,          // 77
  5, 124, 8, 16, 32, 124,         // 78
  5, 56, 68, 68, 68, 56,          // 79
  4, 126, 18, 18, 12,             // 80
  5, 56, 68, 100, 68, 184,        // 81
  4, 124, 20, 52, 88,             // 82
  4, 76, 146, 146, 100,           // 83
  5, 4, 4, 124, 4, 4,             // 84
  5, 60, 64, 64, 64, 60,          // 85
  5, 28, 32, 64, 32, 28,          // 86
  5, 60, 64, 112, 64, 60,         // 87
  6, 132, 72, 48, 48, 72, 132,    // 88
  5, 4, 8, 240, 8, 4,             // 89
  5, 68, 100, 84, 76, 68,         // 90
  2, 62, 34,                      // 91
  5, 2, 4, 8, 16, 32,             // 92
  2, 34, 62,                      // 93
  3, 4, 2, 4,                     // 94
  4, 64, 64, 64, 64,              // 95
  2, 2, 4,                        // 96
  3, 96, 80, 112,                 // 97
  3, 120, 80, 112,                // 98
  3, 48, 72, 72,                  // 99
  3, 112, 80, 120,                // 100
  3, 120, 88, 88,                 // 101
  3, 32, 120, 40,                 // 102
  3, 88, 88, 120,                 // 103
  3, 120, 16, 112,                // 104
  1, 116,                         // 105
  2, 64, 56,                      // 106
  3, 120, 16, 104,                // 107
  1, 124,                         // 108
  5, 96, 16, 32, 16, 96,          // 109
  3, 96, 16, 96,                  // 110
  4, 48, 72, 72, 48,              // 111
  3, 120, 24, 24,                 // 112
  3, 24, 24, 120,                 // 113
  3, 120, 8, 8,                   // 114
  3, 72, 84, 36,                  // 115
  3, 16, 120, 80,                 // 116
  3, 112, 64, 112,                // 117
```

```
  3, 48, 64, 48,                    // 118
  5, 48, 64, 112, 64, 48,           // 119
  4, 72, 48, 48, 72,                // 120
  3, 16, 96, 16,                    // 121
  4, 72, 104, 88, 72,               // 122
  4, 16, 124, 130, 130,             // 123
  1, 255,                           // 124
  4, 130, 130, 124, 16,             // 125
  7, 16, 8, 8, 16, 32, 32, 16,      // 126
  0,                                // 127
  0,                                // 128
  0,                                // 129
  0,                                // 130
  0,                                // 131
  0,                                // 132
  0,                                // 133
  0,                                // 134
  0,                                // 135
  0,                                // 136
  0,                                // 137
  0,                                // 138
  0,                                // 139
  0,                                // 140
  0,                                // 141
  0,                                // 142
  0,                                // 143
  0,                                // 144
  0,                                // 145
  0,                                // 146
  0,                                // 147
  0,                                // 148
  0,                                // 149
  0,                                // 150
  0,                                // 151
  0,                                // 152
  0,                                // 153
  0,                                // 154
  0,                                // 155
  0,                                // 156
  0,                                // 157
  0,                                // 158
  0,                                // 159
  0,                                // 160
  0,                                // 161
  0,                                // 162
  0,                                // 163
  0,                                // 164
  0,                                // 165
```

```
  0,                                  // 166
  0,                                  // 167
  0,                                  // 168
  0,                                  // 169
  0,                                  // 170
  0,                                  // 171
  0,                                  // 172
  0,                                  // 173
  0,                                  // 174
  0,                                  // 175
  1, 4,                               // 176
  0,                                  // 177
  0,                                  // 178
  0,                                  // 179
  0,                                  // 180
  0,                                  // 181
  0,                                  // 182
  0,                                  // 183
  0,                                  // 184
  0,                                  // 185
  2, 0, 0,                            // 186
  0,                                  // 187
  0,                                  // 188
  0,                                  // 189
  0,                                  // 190
  0,                                  // 191
  0,                                  // 192
  0,                                  // 193
  0,                                  // 194
  0,                                  // 195
  0,                                  // 196
  0,                                  // 197
  0,                                  // 198
  0,                                  // 199
  0,                                  // 200
  0,                                  // 201
  0,                                  // 202
  0,                                  // 203
  0,                                  // 204
  0,                                  // 205
  0,                                  // 206
  0,                                  // 207
  0,                                  // 208
  0,                                  // 209
  0,                                  // 210
  0,                                  // 211
  0,                                  // 212
  0,                                  // 213
```

```
  0,                                  // 214
  0,                                  // 215
  0,                                  // 216
  0,                                  // 217
  0,                                  // 218
  0,                                  // 219
  0,                                  // 220
  0,                                  // 221
  0,                                  // 222
  0,                                  // 223
  0,                                  // 224
  0,                                  // 225
  0,                                  // 226
  0,                                  // 227
  0,                                  // 228
  0,                                  // 229
  0,                                  // 230
  0,                                  // 231
  0,                                  // 232
  0,                                  // 233
  0,                                  // 234
  0,                                  // 235
  0,                                  // 236
  0,                                  // 237
  0,                                  // 238
  0,                                  // 239
  0,                                  // 240
  0,                                  // 241
  0,                                  // 242
  0,                                  // 243
  0,                                  // 244
  0,                                  // 245
  0,                                  // 246
  0,                                  // 247
  0,                                  // 248
  0,                                  // 249
  0,                                  // 250
  0,                                  // 251
  0,                                  // 252
  0,                                  // 253
  0,                                  // 254
  0,                                  // 255
};

char c = keyboard.read();
bool firstAlarm = false;
bool secondAlarm = false;
bool thirdAlarm = false;
```

```
bool noAlarm = false;

// unsigned long startTime;
// const unsigned long duration = 5000;

#define PS2_F1 49
#define PS2_F2 50
#define PS2_F3 51
#define PS2_F4 52
#define PS2_F5 53
#define PS2_F6 54

void setup() {
  Serial.begin(9600);
  // startTime = millis();
  keyboard.begin(DATAPIN, IRQPIN);
  pinMode(BUZZER, OUTPUT);
  myRTC.begin();
  P.begin();
  P.setFont(newFont);


  // Parameters

  //   alarmType: A value from the ALARM_TYPES_t enumeration, above.
(ALARM_TYPES_t)
  //   seconds: The seconds value to set the alarm to. (uint8_t)
  //   minutes: The minutes value to set the alarm to. (uint8_t)
  //   hours: The hours value to set the alarm to. (uint8_t)
  //   dayOrDate: The day of the week or the date of the month. For
day of the week,
  //   use a value from the Time library timeDayOfWeek_t
enumeration,
  //   i.e. dowSunday, dowMonday, dowTuesday, dowWednesday,
dowThursday, dowFriday, dowSaturday. (uint8_t)

  // Values for Alarm 1

  //   ALM1_EVERY_SECOND -- causes an alarm once per second.
  //   ALM1_MATCH_SECONDS -- causes an alarm when the seconds match
(i.e. once per minute).
  //   ALM1_MATCH_MINUTES -- causes an alarm when the minutes and
seconds match.
  //   ALM1_MATCH_HOURS -- causes an alarm when the hours and
minutes and seconds match.
  //   ALM1_MATCH_DATE -- causes an alarm when the date of the month
and hours and minutes and seconds match.
  //   ALM1_MATCH_DAY -- causes an alarm when the day of the week
and hours and minutes and seconds match.
```

```
  // Syntax
  //   myRTC.setAlarm(alarmType, seconds, minutes, hours,
dayOrDate);

  // // second
  // myRTC.setAlarm(DS3232RTC::ALM1_EVERY_SECOND, 0, 0, 0, 1);
  // // seconds
  // myRTC.setAlarm(DS3232RTC::ALM1_MATCH_SECONDS, 3, 0, 0, 1);
  // // minutes
  // myRTC.setAlarm(DS3232RTC::ALM1_MATCH_MINUTES, 5, 30, 0, 1);
  // // hours
  // myRTC.setAlarm(DS3232RTC::ALM1_MATCH_HOURS, 3, 3, 3, 1);
  // // date
  // myRTC.setAlarm(DS3232RTC::ALM1_MATCH_DATE, 3, 3, 3, 3);
  // // day
  // myRTC.setAlarm(DS3232RTC::ALM1_MATCH_DAY, 3, 3, 3,
dowWednesday);


  // // set alarm by seconds every minute
  // // set Alarm 1 to occur at 5 seconds after every minute
  // myRTC.setAlarm(DS3232RTC::ALM1_MATCH_SECONDS, 3, 0, 0, 1);


  // clear the alarm flag
  // myRTC.alarm(DS3232RTC::ALARM_1);

  // setSyncProvider() causes the Time library to synchronize with
the
  // external RTC by calling RTC.get() every five minutes by
default.
  setSyncProvider(myRTC.get);
  // Serial << F("RTC Sync");
  // if (timeStatus() != timeSet)
  //   Serial << F(" FAIL!");
  // Serial << endl;

  P.displayClear();
}

void loop() {
  // change time_t (uint32_t) to uint16_t if need to free more
memory
  static time_t tLast;
  time_t t;
  tmElements_t tm;

  // keyboard input
```

```
  if (keyboard.available()) {
    char c = keyboard.read();
    // check for some of the special keys
    if (c == PS2_ENTER) {
      Serial.println();
    } else if (c == PS2_TAB) {
      Serial.print("Tab");
    } else if (c == PS2_ESC) {
      Serial.print("ESC");
    } else if (c == PS2_PAGEDOWN) {
      Serial.print("PgDn");
    } else if (c == PS2_PAGEUP) {
      Serial.print("PgUp");
    } else if (c == PS2_LEFTARROW) {
      Serial.print("Left");
    } else if (c == PS2_RIGHTARROW) {
      Serial.print("Right");
    } else if (c == PS2_UPARROW) {
      Serial.print("Up");
    } else if (c == PS2_DOWNARROW) {
      Serial.print("Down");
    } else if (c == PS2_DELETE) {
      Serial.print("Del");
    } else if (c == 33) {  // !
      // second
      myRTC.setAlarm(DS3232RTC::ALM1_EVERY_SECOND, 0, 0, 0, 1);
      Serial << "set alarm 1";
      firstAlarm = true;
      secondAlarm = false;
      thirdAlarm = false;
      noAlarm = false;
    } else if (c == 64) {  // @
      // seconds
      myRTC.setAlarm(DS3232RTC::ALM1_MATCH_SECONDS, 3, 0, 0, 1);
      Serial << "set alarm 2";
      firstAlarm = false;
      secondAlarm = true;
      thirdAlarm = false;
      noAlarm = false;
    } else if (c == 35) {  // #
      // minutes
      myRTC.setAlarm(DS3232RTC::ALM1_MATCH_MINUTES, 5, 30, 0, 1);
      Serial << "set alarm 3";
      firstAlarm = false;
      secondAlarm = false;
      thirdAlarm = true;
      noAlarm = false;
    } else if (c == 36) {  // enables alarm with "$" in keyboard
```

```
      // hours
      myRTC.setAlarm(DS3232RTC::ALM1_MATCH_HOURS, 3, 3, 3, 1);
      digitalWrite(BUZZER, HIGH);
    } else if (c == 37) {  // enables alarm with "%" in keyboard
      // date
      myRTC.setAlarm(DS3232RTC::ALM1_MATCH_DATE, 3, 3, 3, 3);
      digitalWrite(BUZZER, HIGH);
    } else if (c == 94) {  // enables alarm with "^" in keyboard
      // day
      myRTC.setAlarm(DS3232RTC::ALM1_MATCH_DAY, 3, 3, 3,
dowWednesday);
      digitalWrite(BUZZER, HIGH);
    } else if (c == 49 && firstAlarm == true) {  // disables alarm
with "1" in keyboard
      digitalWrite(BUZZER, LOW);
      firstAlarm = false;
      noAlarm = true;
      myRTC.alarm(DS3232RTC::ALARM_1);
      myRTC.clearAlarm(DS3232RTC::ALARM_1);
    } else if (c == 50 && secondAlarm == true) {  // disables alarm
with "2" in keyboard
      digitalWrite(BUZZER, LOW);
      secondAlarm = false;
      noAlarm = true;
      myRTC.alarm(DS3232RTC::ALARM_1);
      myRTC.clearAlarm(DS3232RTC::ALARM_1);
    } else if (c == 51 && thirdAlarm == true) {  // disables alarm
with "3" in keyboard
      digitalWrite(BUZZER, LOW);
      thirdAlarm = false;
      noAlarm = true;
      myRTC.alarm(DS3232RTC::ALARM_1);
      myRTC.clearAlarm(DS3232RTC::ALARM_1);
    } else if (c == 38) {
      firstAlarm = false;
      secondAlarm = false;
      thirdAlarm = false;
      noAlarm = true;
      myRTC.alarm(DS3232RTC::ALARM_1);
      myRTC.clearAlarm(DS3232RTC::ALARM_1);
      digitalWrite(BUZZER, LOW);
    } else {
      // otherwise, just print all normal characters
      Serial.print(c);
    }
  }
```

```
  // check for input to set the RTC, minimum length is 12, i.e.
yy,m,d,h,m,s
  if (Serial.available() >= 12) {
    // note that the tmElements_t Year member is an offset from
1970,
    // but the RTC wants the last two digits of the calendar year.
    // use the convenience macros from the Time Library to do the
conversions.

    int y = Serial.parseInt();
    if (y >= 100 && y < 1000)
      Serial << F("Error: Year must be two digits or four digits!")
<< endl;
    else {
      if (y >= 1000)
        tm.Year = CalendarYrToTm(y);
      else  // (y < 100)
        tm.Year = y2kYearToTm(y);
      tm.Month = Serial.parseInt();
      tm.Day = Serial.parseInt();
      tm.Hour = Serial.parseInt();
      tm.Minute = Serial.parseInt();
      tm.Second = Serial.parseInt();
      t = makeTime(tm);
      myRTC.set(t);  // use the time_t value to ensure correct
weekday is set
      setTime(t);
      Serial << F("RTC set to: ");
      // printDateTime(t);

      Serial << endl;
      // dump any extraneous input
      while (Serial.available() > 0)
        Serial.read();
    }
  }

  t = now();

  if (t != tLast) {
    tLast = t;

    // P.setIntensity(0);
    // changes brightness according to light sensor
    if (analogRead(LIGHT_SENSOR) < 100) {
      // terang banget
      P.setIntensity(10);
    } else if (analogRead(LIGHT_SENSOR) < 200) {
```

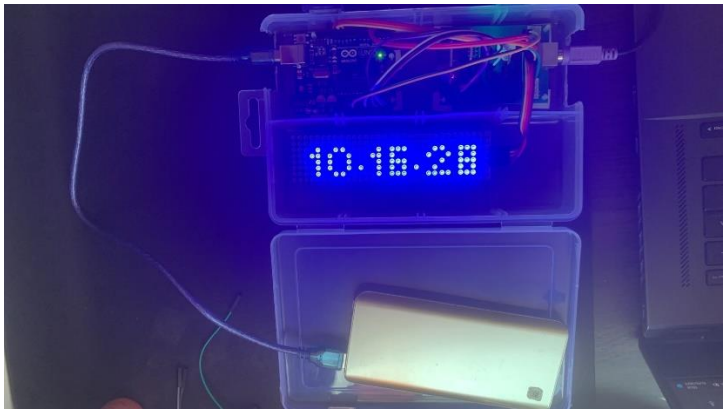
```
      // terang
      P.setIntensity(8);
    } else if (analogRead(LIGHT_SENSOR) < 500) {
      // normal
      P.setIntensity(5);
    } else if (analogRead(LIGHT_SENSOR) < 800) {
      // gelap
      P.setIntensity(3);
    } else {
      // gelap banget
      P.setIntensity(0);
    }

    // // prints calendar and clock
    // printDateTime(t);
    // // prints LM35, RTC Temp, and LDR, current display on dot
matrix
    // Serial << "LM35: " << lm35.getTemp(CELCIUS) << " "
    //        << "RTC: " << myRTC.temperature() / 4.0 << " LDR:" <<
analogRead(LIGHT_SENSOR) << " " << buffer;
    uint16_t celciusTemp = lm35.getTemp(CELCIUS);

    Serial << buffer << " " << celciusTemp;


    if (firstAlarm == true && myRTC.checkAlarm(DS3232RTC::ALARM_1))
{
      Serial << " ALARM 1";
      snprintf(buffer, sizeof(buffer), "1. 5024201073");
      digitalWrite(BUZZER, HIGH);
    } else if (secondAlarm == true &&
myRTC.checkAlarm(DS3232RTC::ALARM_1)) {
      Serial << " ALARM 2";
      snprintf(buffer, sizeof(buffer), "2. 5024201073 Iqbal
Muchlis");
      digitalWrite(BUZZER, HIGH);

    } else if (thirdAlarm == true &&
myRTC.checkAlarm(DS3232RTC::ALARM_1)) {
      Serial << " ALARM 3";
      snprintf(buffer, sizeof(buffer), "3. Keyboard Input");
      digitalWrite(BUZZER, HIGH);
    } else if (noAlarm == true) {
      // snprintf(buffer, sizeof(buffer), " ");
      if (second(t) == 9 || second(t) == 39 || second(t) == 14 ||
second(t) == 44) {
        // myRTC.clearAlarm(DS3232RTC::ALARM_1);
        snprintf(buffer, sizeof(buffer), " ");
```

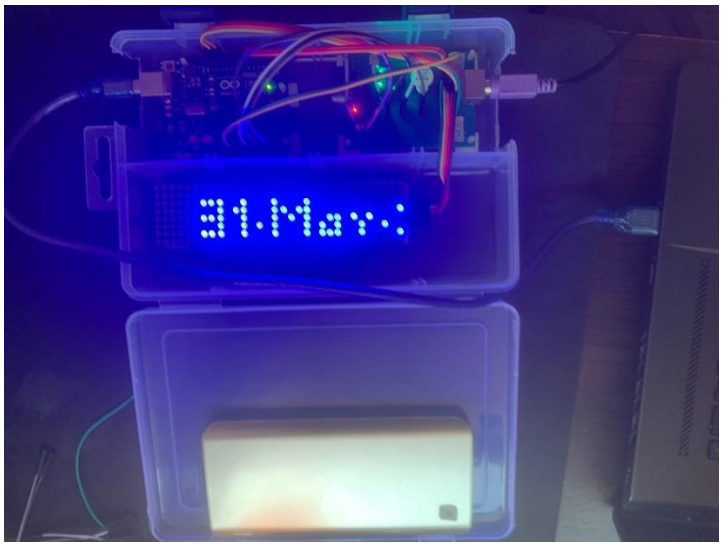
```
      } else if ((second(t) >= 10 && second(t) <= 13) || (second(t)
>= 40 && second(t) <= 43)) {
        // displays calendar for 3 seconds
        digitalWrite(BUZZER, LOW);
        snprintf(buffer, sizeof(buffer), "%d.%s.%d", day(t),
monthShortStr(month(t)), _DEC(year(t)));
        Serial << " CALENDAR";
      } else if ((second(t) >= 14 && second(t) <= 17) || (second(t)
>= 44 && second(t) <= 46)) {
        // displays temperature for 3 seconds
        digitalWrite(BUZZER, LOW);
        snprintf(buffer, sizeof(buffer), "%d °C", celciusTemp);
        Serial << " TEMP";
      } else {
        // displays clock as a default
        digitalWrite(BUZZER, LOW);
        snprintf(buffer, sizeof(buffer), "%d.%d.%d", hour(t),
minute(t), second(t));
      }
    }

    Serial << endl;
  }

  if (P.displayAnimate()) {
    if (firstAlarm == true || secondAlarm == true || thirdAlarm ==
true || noAlarm == false) {
      P.displayText(buffer, PA_RIGHT, 60, 60, PA_SCROLL_LEFT,
PA_SCROLL_LEFT);
    }
    if (second(t) == 9 || second(t) == 39 || second(t) == 14 ||
second(t) == 44) {
      // dont change PA_OPENING, it fixes weird calendar + temp
scrolling
      P.displayText(buffer, PA_CENTER, 60, 60, PA_OPENING,
PA_OPENING);
    } else if (((second(t) >= 10 && second(t) <= 13) || (second(t)
>= 40 && second(t) <= 43)) || ((second(t) >= 14 && second(t) <= 16)
|| (second(t) >= 44 && second(t) <= 45))) {
      P.displayText(buffer, PA_RIGHT, 60, 60, PA_SCROLL_LEFT,
PA_SCROLL_LEFT);
    } else {
      P.displayText(buffer, PA_CENTER, 60, 60, PA_NO_EFFECT,
PA_NO_EFFECT);
    }
    P.displayReset();
  }
}
```

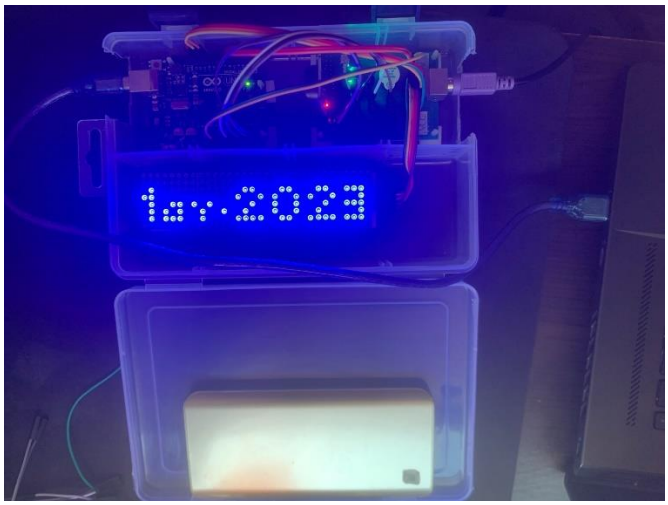
```
  Serial << _DEC(val);
  if (delim > 0) Serial << delim;
  return;
}
```

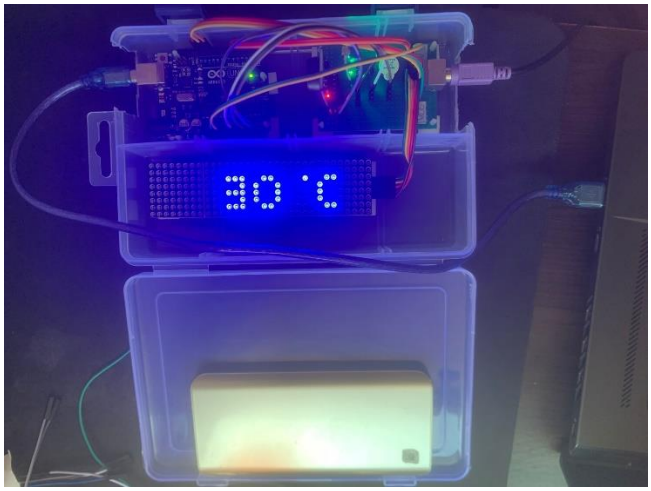
- Hasil display

    - clock, yang dimana pada saat detik 10, 13 dan 40, 43 akan tampil kalender dan suhu

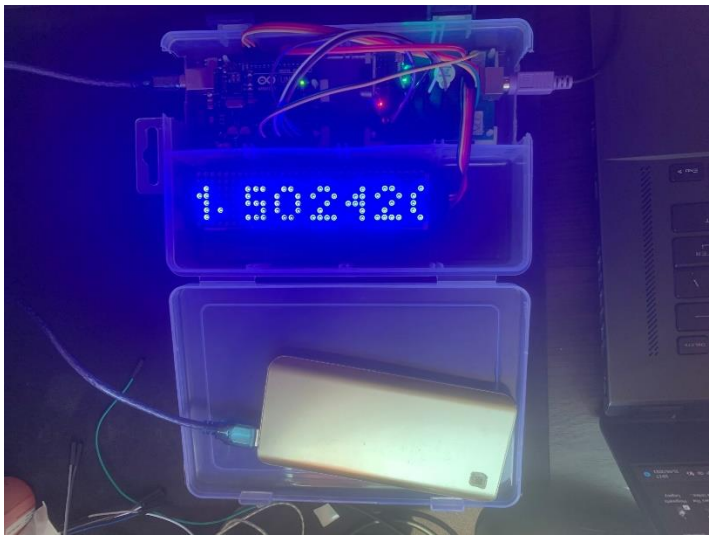




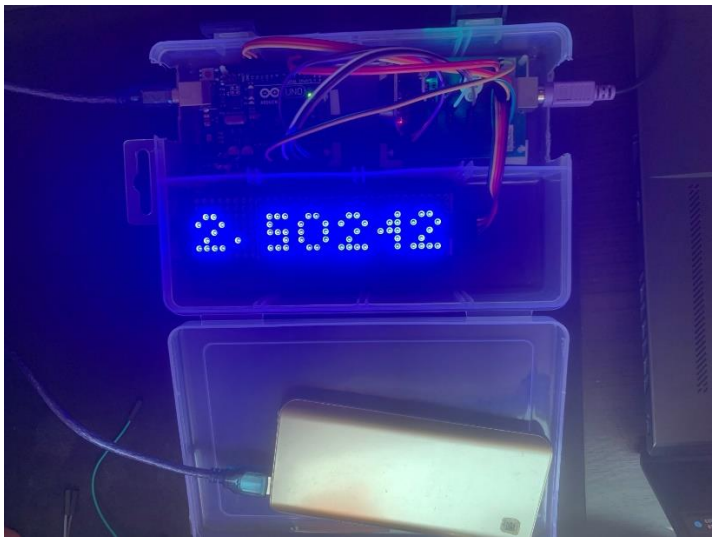


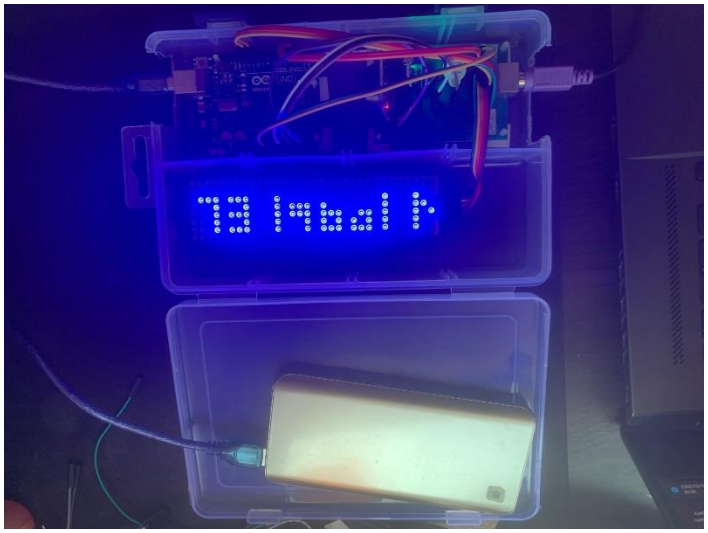


- display pada saat alarm 1 bernyala

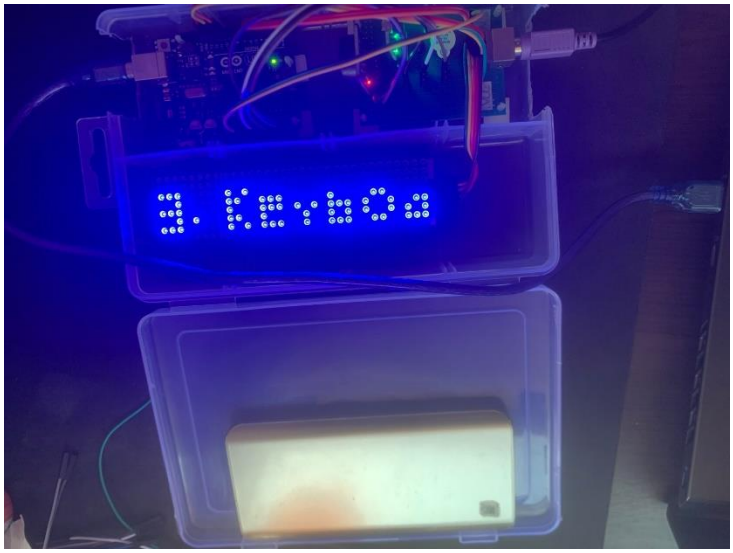


- display pada saat alarm 2 bernyala





- display pada saat alarm 3 bernyala



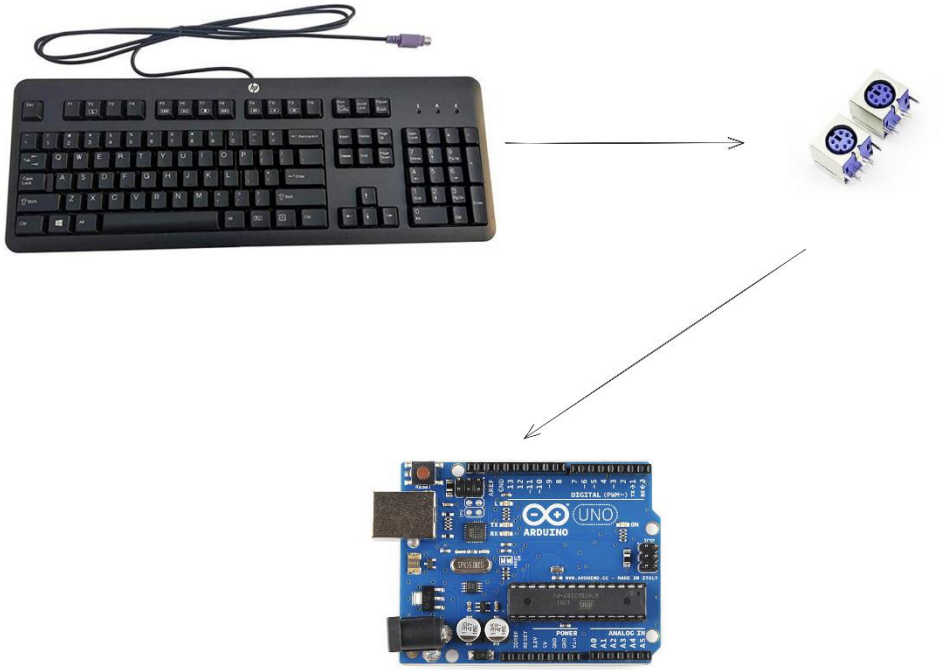
V. KESIMPULAN

Kode saya memiliki kekurangan yaitu tidak dapat:

- tidak dapat setting detail alarm (contoh: durasi alarmnya) dengan keyboard.

- hanya bisa menampilkan satu per satu karakter (contoh: saat tekan 1 pada keyboard akan tampil 1 tapi saat tekan 2 yang seharusnya menampilkan 12 tetapi hanya 2 saja) pada dot-matrix dengan keyboard.

Detail rancangan input keyboard:



Dengan menggunakan ps/2 keyboard, saya dapat memakai female ps/2 socket sehingga dapat digunakannya pada arduino. Untuk implementasinya kodingannya, saya memakai sebuah library untuk mengubah scan code dari ps/2 menjadi ascii character.