**Kaderisasi Workshop HME 2017**

**Tugas Kelompok 1 *“WSCalc 2017”***

1. **Data Kelompok**

*Isi tabel berikut dengan anggota kelompok anda!*

Kelompok :

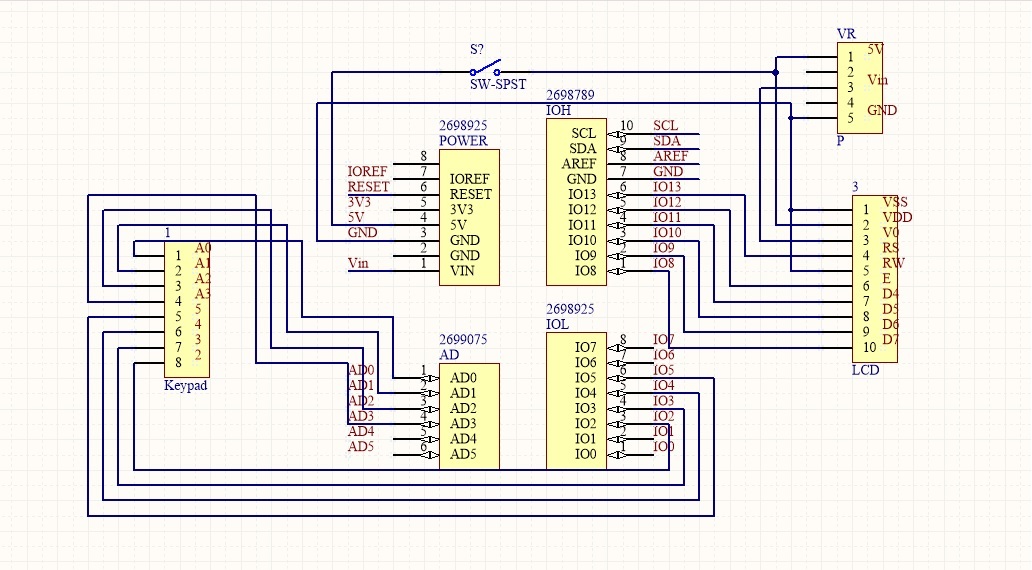
|  |  |  |
| --- | --- | --- |
| No | Nama | NIM |
| 1 | Isro Syaeful Iman | 13116058 |
| 2 | Sandy Gunawan | 13216038 |
| 3 | Thariq Ramadhan | 13216096 |
| 4 | Sayyid Irsyadul Ibad | 13215068 |
| 5 | Gabriel Dicky A | 18316027 |
| 6 |  |  |

1. **PCB Designing dan Arduino**

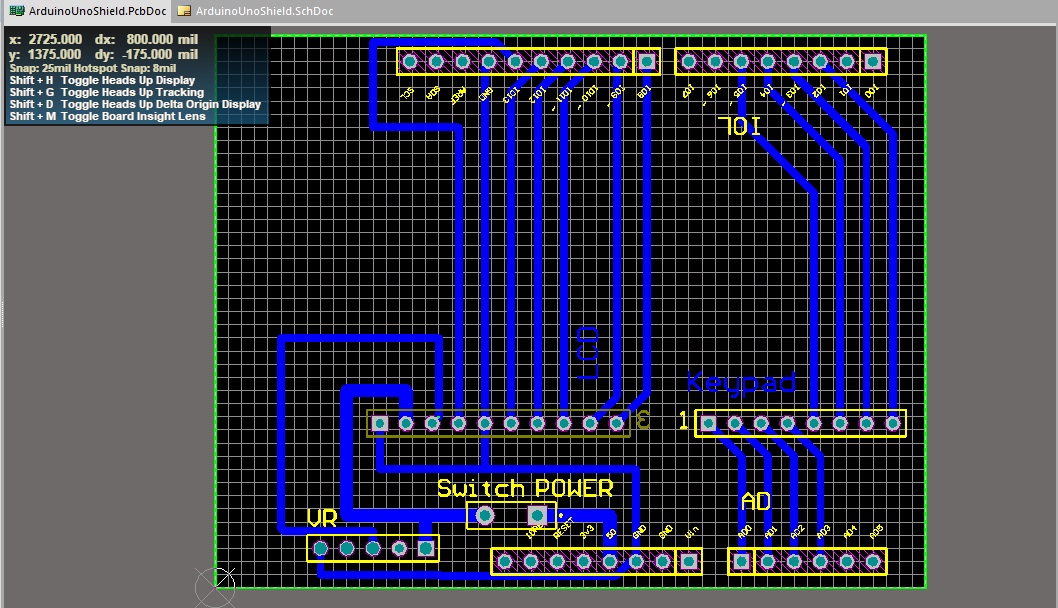
*Isi bagian ini dengan hal-hal yang anda dapatkan dalam melakukan poject WSCalc 2017 ini. Boleh menambahkan review dari Day 1 dan Day 2, dan hal-hal yang anda temukan dan pelajari selama mengerjakan project ini. Dapat diisi dengan poin-poin ataupun bentukan paragraf.*

Dalam melakukan *project* WSCalc 2017 ini, kelompok kami mendapatkan beberapa hal-hal yang menarik sekaligus penting dalam elektro praktis untuk sekarang ataupun nantinya. Hal-hal tersebut seperti mendesain PCB, serta melakukan pemrograman dengan menggunakan mikro kontroler Arduino/AVR. Hal tersebut menjadi penting dikarenakan dalam mengerjakan hal-hal berbasis elektro praktis tentu terkadang kita akan menghadapi keadaan di mana kita akan kesulitan dalam membuat rangkaian apabila kita harus membuatnya secara tradisional, secara masif misalnya, ataupun adanya tuntutan untuk membuat suatu rangkaian dalam ukuran yang lebih kecil, di mana tentu akan sangat sulit apa bila kita harus mengerjakannya secara manual. Selain itu, dewasa ini kita sering dihadapkan dengan tuntutan untuk membuat *device* (perangkat) berbasis *Internet of Things* (*IoT*) yang di dalamnya tentu diperlukan suatu komponen yang “cerdas” dan dapat mengontrol kerja dari rangkaian tersebut. Dari tuntutan itu, kita memerlukan adanya pendesainan PCB agar dapat menghemat waktu pengerjaan rangkaian yang jumlahnya masif, serta membantu menyelesaikan masalah dalam meminimalisasi ruang yang diperlukan untuk rangkaian. Pada *project* ini piranti lunak PCB *designing* yang digunakan adalah KiCad. Kenapa KiCad? KiCad dipilih karena dalam penggunaannya, KiCad tidak memerlukan lisensi khusus untuk menggunakannya secara komersil, sedangkan yang lainnya jika ingin digunakan secara komersil memerlukan pembelian lisensi. Akan tetapi, hal tersebut baru menyelesaikan tuntutan pertama. Oleh karena itu, untuk menyelesaikan tuntutan kedua, kita memerlukan *skill* dalam melakukan pemrograman pada *microcontroller* yang pada *project* ini adalah Arduino/AVR yang merupakan *microcontroller* yang cukup murah dan tergolong mudah untuk diprogram.

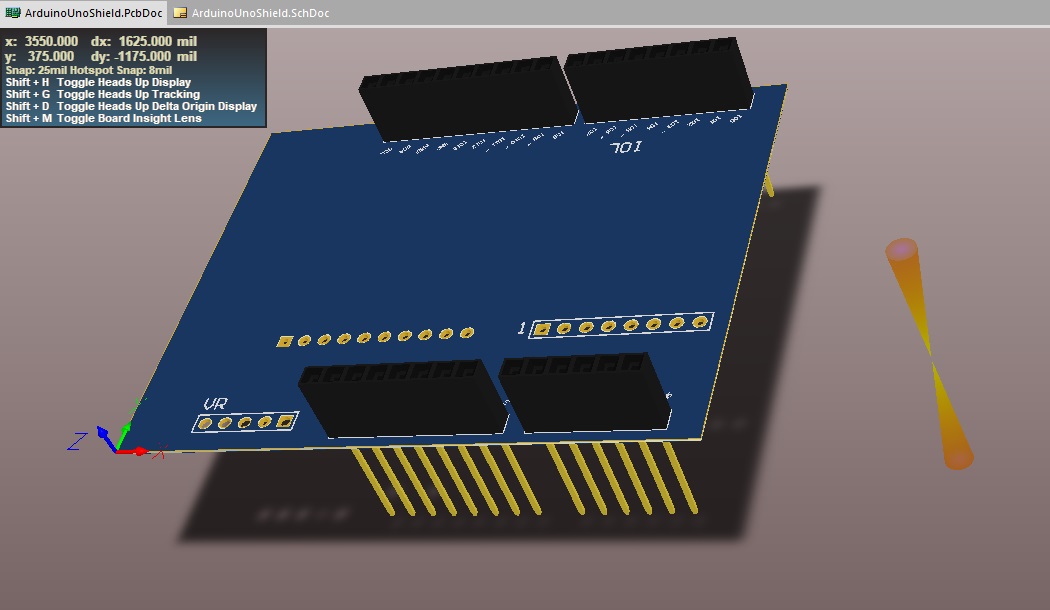
1. **Dokumentasi Pengerjaan**
2. Desain pcb di Altium.
3. Schematic



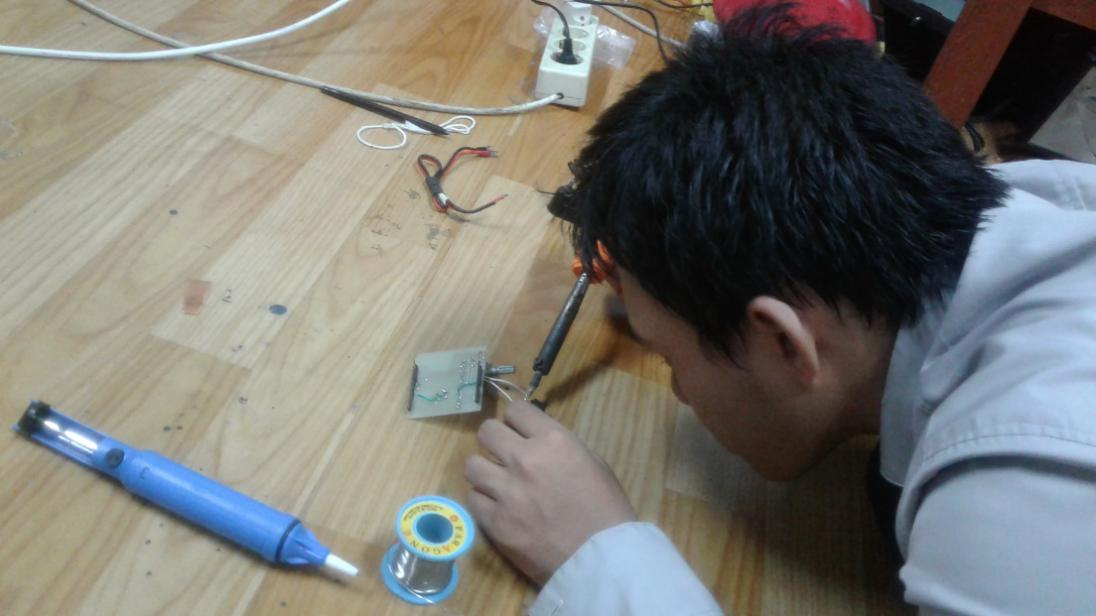
1. Board View (2D)



1. 3D view



1. Menyolder header-header pada pcb



1. Source code pada Arduino IDE

#include <LiquidCrystal.h>

#include <Key.h>

#include <Keypad.h>

#include <math.h>

LiquidCrystal lcd(13,12,11,10,9,8);

int sum;

float sumfl;

int button1=6;

int i;

int A=0;

int B=0;

int intA;

int intB;

String str="";

String sumstr="";

String tempchar;

const byte ROWS = 4; //four rows

const byte COLS = 4; //four columns

char keys[ROWS][COLS] =

{

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'}

};

byte rowPins[ROWS] = {A0,A1,A2,A3}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {5,4,3,2}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

bool cekmode=false;

void setup(){

Serial.begin(9600);

pinMode(button1, INPUT\_PULLUP);

}

void loop(){

cekmode=false;

char key = keypad.getKey();

if (digitalRead(button1)==LOW){ // ketika tdk mendeteksi inut pullup dari button

delay(500);

B=B+1;

B=B%5;

lcd.clear();

}

switch (B) {

case 0:

lcd.setCursor(0, 0);

lcd.print("WSCALCULATOR"); //welcome screen

delay(2000);

for (i = 0 ; i < 16; i ++) {

lcd.scrollDisplayRight();

if (digitalRead(button1)==LOW){ //jika input pullup = low, running text jalan

delay(500);B=B+1;

lcd.clear();

lcd.setCursor(0,0);

break;}

delay(700);

}

break;

case 1:

while (not(cekmode)) {

key = keypad.getKey();

tempchar=tempchar+key;

if ((key)&& (tempchar!="\*" && tempchar!="#" && tempchar!="A" && tempchar!="B" && tempchar!="C" && tempchar!="D")){ // selama tdk ada simbol, maka membaca input dari keypad

Serial.println(key);

lcd.print(key);

delay(200);

str=str+key;

}

if(digitalRead(button1)==LOW){

delay(500);

cekmode=true;

B=B+1;

lcd.clear();

lcd.setCursor(0,0);

intA=str.toInt();

str="";

Serial.println(intA);//menampilkan input angka ke lcd

}

}

break;

case 2:

while(not(cekmode)){

key=keypad.getKey();

tempchar="";

if (key){

tempchar=tempchar+key;

delay(300);

Serial.println(tempchar);

if (tempchar=="#") {

A =A+1;

A = A % 13 ;

Serial.println(A);

tempchar="";

} else if (tempchar=="\*") {

A =A-1;

A = A % 13 ;

Serial.println(A);

tempchar="";

}

switch (A) { // print untu setiap operator

case 1:

lcd.print("+ ");

break;

case 2:

lcd.print("- ");

break;

case 3:

lcd.print("X ");

break;

case 4 :

lcd.print("sin ");

break;

case 5 :

lcd.print("cos ");

break;

case 6 :

lcd.print("tan " );

break;

case 7:

lcd.print("sqrt ");

break;

case 8:

lcd.print("power ");

break;

case 9:

lcd.print("Binary ");

break;

case 10:

lcd.print("Hexadecimal ");

break;

case 11:

lcd.print("Log ");

break;

case 12:

lcd.print("Exp ");

break;

case -12:

lcd.print("+ ");

break;

case -11:

lcd.print("- ");

break;

case -10:

lcd.print("X ");

break;

case -9 :

lcd.print("sin ");

break;

case -8 :

lcd.print("cos ");

break;

case -7 :

lcd.print("tan ");

break;

case -6:

lcd.print("sqrt ");

break;

case -5:

lcd.print("power ");

break;

case -4:

lcd.print("Binary ");

break;

case -3:

lcd.print("Hexadecimal ");

break;

case -2:

lcd.print("Logaritma ");

break;

case -1:

lcd.print("exponen ");

break;

case 0:

lcd.print("/ ");

break;

}

lcd.setCursor(0,0);

}

if(digitalRead(button1)==LOW){

delay(500);

cekmode=true;

B=B+1;

lcd.clear();

lcd.setCursor(0,0);

Serial.println(intA);

Serial.println(A);

}

}

break;

case 3:

while (not(cekmode)) {

key = keypad.getKey();

if (key){

Serial.println(key);

lcd.print(key);

delay(200);

str=str+key; //menerima input angka kedua dari keypad

}

if(digitalRead(button1)==LOW){

delay(500);

cekmode=true;

B=B+1;

lcd.clear();

lcd.setCursor(0,0);

intB=str.toInt();

str="";

Serial.println(intB); // menampilkan angka kedua di layar lcd

}

}

break;

case 4:

while (not(cekmode)){

lcd.setCursor(1,0);

switch (A) {

case 1:

sum=intA+intB; // untuk penjumlahan

lcd.print(sum);

delay(500);

break;

case 2:

sum=intA-intB; //untuk pengurangan

lcd.print(sum);

delay(500);

break;

case 3:

sum=intA\*intB; //untuk perkalian

lcd.print(sum);

delay(500);

break;

case 4:

sum=intA\*sin(intB\*PI/180); // untuk sin

lcd.print(sum);

delay(500);

break;

case 5:

sum=intA\*cos(intB\*PI/180); //untuk cos

lcd.print(sum);

delay(500);

break;

case 6:

sum=intA\*tan(intB\*PI/180); //untuk tangen

lcd.print(sum);

delay(500);

break;

case 7:

sum=intA\*sqrt(intB); //untuk akar

lcd.print(sum);

delay(500);

break;

case 8:

sum=pow(intA,intB); //untuk kuadrat

lcd.print(sum);

delay(500);

break;

case 9:

sumstr=String(intB, BIN); //untuk biner

lcd.print(sumstr);

delay(500);

break;

case 10:

sumstr=String(intB, HEX); //untuk hexadecimal

lcd.print(sum);

delay(500);

break;

case 11:

sum=intA\*log10(intB); //untuk logaritma

lcd.print(sum);

delay(500);

break;

case 12:

sum=intA\*exp(intB); //untuk exponential

lcd.print(sum);

delay(500);

break;

case -12:

sum=intA+intB; //untuk angka negatif

lcd.print(sum);

delay(500);

break;

case -11:

sum=intA-intB; //untuk angka negatif

lcd.print(sum);

delay(500);

break;

case -10:

sum=intA\*intB;

lcd.print(sum);

delay(500);

break;

case -9:

sum=intA\*sin(intB\*PI/180);

lcd.print(sum);

delay(500);

break;

case -8:

sum=intA\*cos(intB\*PI/180);

lcd.print(sum);

delay(500);

break;

case -7:

sum=intA\*tan(intB\*PI/180);

lcd.print(sum);

delay(500);

break;

case -6:

sum=intA\*sqrt(intB);

lcd.print(sum);

delay(500);

break;

case -5:

sum=pow(intA,intB);

lcd.print(sum);

delay(500);

break;

case -4:

sumstr=String(intB, BIN);

lcd.print(sumstr);

delay(500);

break;

case -3:

sumstr=String(intB, HEX);

lcd.print(sum);

delay(500);

break;

case -2:

sum=intA\*log10(intB);

lcd.print(sum);

delay(500);

break;

case -1:

sum=intA\*exp(intB);

lcd.print(sum);

delay(500);

break;

case 0:

sumfl:intA/intB;

lcd.print(sumfl);

delay(500);

break;

}

if(digitalRead(button1)==LOW){

delay(500);

cekmode=true;

B=0;

lcd.clear();

lcd.print("WSCALCULATOR"); //back to welcome screen

lcd.setCursor(0,0);

str="";

}

}

break;

}

delay(10);

}

1. Testing menggunakan Arduino yang dilengkapi shield.



1. **Keterangan Tambahan**

Pertama, kami melakukan testing untuk keypad dan lcd dulu pada media breadboard. Kami mencoba keduanya secara terpisah terlebih dahulu. Jadi, kami melakukan testing untuk keypad dengan menggunakan serial monitor untuk mengecek dulu apakah keypad bisa berjalan untuk menampilkan char dengan Serial.println. Kemudian, kami juga mengetes lcd. Kami membuat running text dulu dan menampilkannya di lcd. Setelah itu, kami menggabungkan keduanya dan mencobanya pada rangkaian menggunakan breadboard. Kami mencoba untuk menangkap input dari keypad, lalu menampilkannya di layar lcd.

Setelah testing dengan breadboard selesai, kami lalu membuat desain untuk pcb sebagai shield arduinonya. Dimensinya adalah 2700 mil x 2100 mil. Secara tata letaknya juga cukup sederhana. Kami membuat 8 lubang header untuk keypad, dan 10 lubang header. Kami meletakkan lubang untuk keypad di sebelah kanan (dilihat dari top view). Sementara lubang untuk lcd kami letakkan di sebelah kiri. Posisi relatifnya linear agar simpel dan mudah untuk dibaca. Sebenarnya untuk lcd sendiri ada 14 header, tapi kami membuat 10 lubang agar irit header karena masalah ekonomi, hehe (meskipun anntinya diperlukan tambahan jumper dulu akhirnya dari header female menuju lcd). Kemudian kami juga menambahkan lubang untuk potensiometer. Potensiometer ini fungsinya untuk mengatur contrast lcd. Desain dilakukan di Altium. Pertama membuat schematic, kemudian di-compile, kemudian mengatur tata letak komponen-komponen di board mode. Setelah itu routing sebagai jalur tembaganya. Lalu cek eror dulu, setelah tidak ada error , tahap terakhir adalh di-print dan dicetak di tempat cetak pcb, Spectra.

Selagi menunggu print pcb, kami juga melanjutkan membuat source code untuk kalkulatornya. Kami memprogramnya di Arduino IDE. Source code Arduino IDE nya bisa dilihat di dokumentasi pengerjaan.

Kemudian setelah shield jadi, kami melakukan simulasi dengan keypad + lcd menggunakan shield Arduino yang sudah dicetak. Mengecek operasi bilangannya. Selesai deh.

Adapun kami juga menemui beberapa kendala ketika mengerjakan tugas kalkulator ini. Pertama, kami lupa membuat lubang untuk pull up button, sehingga setelah di-print kami membuat lubang secara manual dengan dibor untuk ditambahkan ke pcb yang sudah diprint tadi. Lalu, karena keteledoran, lcd kami mati 1 biji. Penyebabnya karena tertukar untuk menyambungkan ke pin VCC dan GND, jadi lcd nya rusak ☹. Ganti deh dengan lcd yang baru.

1. **Pertanyaan**
2. Apa itu internal pull up atau pull down Arduino?

Di dalam Arduino dapat dibilang sudah terdapat sebuah resistor yang besarnya kira-kira 20k-50kΩ di setiap *pin*-nya. Dengan memberikan perintah/menggunakan mode *pull up*, maka dapat diasumsikan kita memasang resistor tersebut. Sedangkan, apabila kita menggunakan mode *pull down*, dapat diasumsikan kita melepas resistor tersebut. Apabila kita menggunakan mode *pull up* dan menghubungkan pin tersebut dengan sebuah *switch* yang terhubung ke *ground*, maka yang terjadi adalah bacaan masukan yang diterima oleh *pin* adalah low (0), karena arus mengalir ke *ground*. Hal ini lebih dipilih dibandingkan menggunakan masukan berupa *high* (1), karena adanya perbedaan kondisi *high* pada Arduino, yaitu 5V, sedangkan pada rangkaian umumnya 3,3V. Akan tetapi untuk *ground*, nilainya tetap sama, yaitu 0V. Sehingga apabila kondisi yang ingin dibaca Arduino adalah *high*, ada kemungkinan bahwa masukan justru dibaca *low.*

1. Jelaskan secara singkat cara kerja WSCalc 2017 Anda!

WSCalc 2017 buatan kelompok kami memiliki sebuah tombol pada PCB serta tombol pada *keypad*. Pada saat pertama kali dinyalakan, layar akan menampilkan sebuah *running text* “WSCALCULATOR” yang berjalan ke kanan sebanyak 16 kali. Selama tombol *keypad* A, B, C, D, \*, serta # tidak ditekan, WSCalc 2017 akan terus membaca input dari *keypad*. Lalu apabila simbol ditekan, WSCalc 2017 akan berhenti membaca masukan angka. Lalu kita dapat menekan angka # ataupun \* untuk memilih operasi hitung yang diinginkan. Lalu, WSCalc 2017 akan meminta lagi masukan untuk angka kedua dari *keypad*. Lalu setelah itu, WSCalc akan menampilkan hasil perhitungan sesuai dengan operasi yang terpilih dari masukan # ataupun \* tersebut. Layar akan menghapus hasil perhitungan tersebut dan akan menampilkan kembali *running text* “WSCALCULATOR” setelah *button* pada PCB ditekan.

1. Apa produk lain yang dapat Anda buat dengan bermodalkan skill yang telah Anda dapatkan dalam pembuatan WSCalc 2017?

Dengan bermodalkan *skill* yang didapat dalam pembuatan WSCalc 2017 ini, ada beberapa produk lain yang dapat dibuat, seperti *finder* yang menampilkan datadari suatu indeks tertentu hasil masukan pengguna, ataupun seperti alat untuk memesan suatu produk dengan suatu kode tertentu dan jumlah tertentu.

1. Mengapa skill PCB Design dan Arduino/AVR penting bagi cakru?

*Skill* PCB *design* penting bagi CaKru, karena bisa jadi ada permintaan dari *client* untuk membuat suatu rangkaian pada *device* tertentu yang nantinya akan diproduksi secara massal. Tentu apabila hubungan pada rangkaian tersebut harus disolder sendiri akan memakan waktu lama. Akan tetapi, apabila kita menggunakan PCB *design*, hasil rangkaian yang telah dibuat dapat diproduksi secara massal lengkap dengan jalur yang menghubungkan suatu komponen dengan komponen lain, dan yang perlu dilakukan untuk merampungkannya adalah cukup memasang dan menyolder komponen ke tempatnya.

Sedangkan *skill* Arduino/AVR penting, karena dewasa ini sudah banyak sekali piranti-piranti berbasis *Internet* *of Things* (*IoT*) yang dibuat dengan memanfaatkan Arduino/AVR sebagai *controller*-nya. Selain itu, Arduino/AVR juga dapat digunakan untuk membuat suatu piranti “cerdas” dengan melakukan pemrograman yang sesuai dengan fungsi yang ingin diterapkan pada piranti tersebut.

1. Selain PCB Design dan Arduino/AVR, apa yang telah Anda dapatkan pada Season 1 Kaderisasi Workshop HME 2017?

Selain PCB *design* dan Arduino/AVR, saya telah mendapatkan beberapa hal lain, seperti beberapa pengenalan tentang alasan pemilihan aplikasi yang berkaitan dibanding aplikasi lainnya. Selain itu, ada pula hal-hal yang menyangkut teknis, seperti bagaimana cara melakukan pencetakan PCB, serta tempatnya, ataupun cara mencari solusi pada pengerjaan *project* yang melibatkan Arduino/AVR.