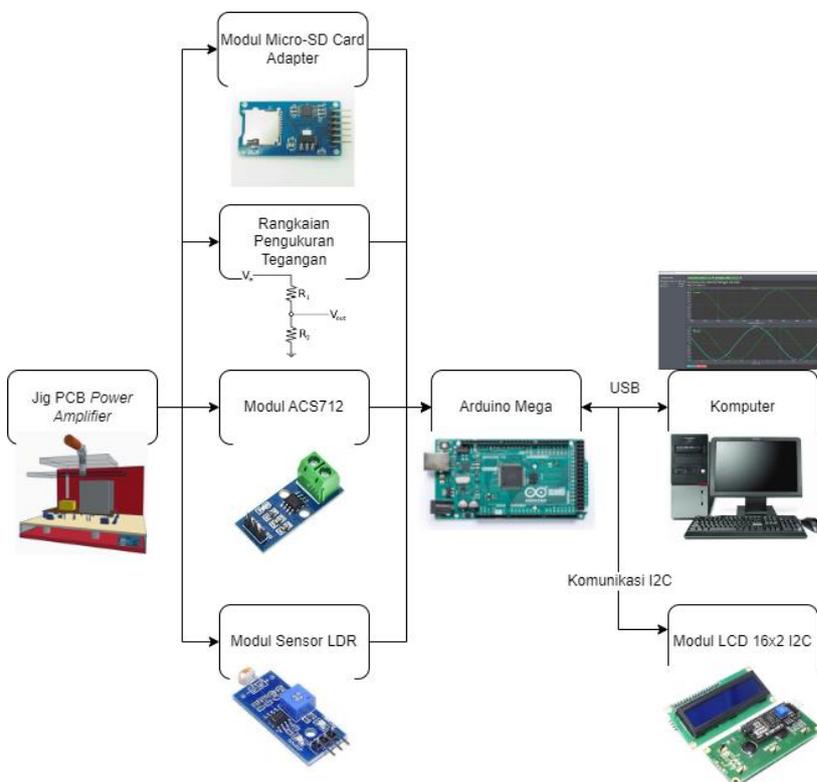


3. PERANCANGAN SISTEM

Bagian ini adalah bagian yang akan menjelaskan semua perancangan dan desain yang digunakan untuk membuat modul uji rangkaian *power amplifier*. Modul uji ini adalah modul yang digunakan untuk melakukan pengetesan *quality control* pada PCB *power amplifier*. Pengetesan yang dilakukan adalah pengukuran tegangan pada beberapa titik di PCB *power amplifier* dan melakukan pembacaan sinyal pada beberapa titik. Modul uji ini terdiri dari beberapa alat yang perlu digunakan yaitu *jig PCB*, *Arduino Mega*, rangkaian pengukuran tegangan, modul ACS712, modul sensor LDR, modul *micro-SD Card adapter*, kabel penghubung, jarum pengukuran, modul LCD 16x2 I2C dan komputer.



Gambar 3.1 Blok Diagram Modul Uji Rangkaian *Power Amplifier* untuk *Quality Control*

Gambar 3.1 adalah blok diagram dari pemasangan *hardware* yang akan digunakan. *Jig PCB* adalah alat yang digunakan sebagai tempat untuk meletakkan PCB *power amplifier* yang akan diuji. Rangkaian pengukuran tegangan adalah rangkaian pembagi tegangan yang berfungsi untuk membatasi tegangan yang masuk pada pin analog *Arduino Mega*. Modul ACS712 adalah modul

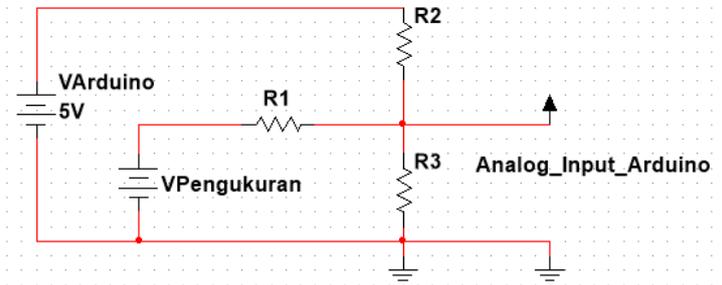
yang berfungsi untuk mengukur arus dan hasil pengukuran disambungkan pada pin analog *Arduino Mega*. Modul sensor LDR adalah modul sensor yang digunakan untuk mendeteksi terdapatnya PCB di *jig* tersebut dimana modul sensor ini disambungkan pada pin digital *Arduino Mega*. Modul *micro-SD Card adapter* adalah modul berfungsi untuk membaca file parameter tegangan maksimal dan minimal untuk *quality control* yang tersimpan di dalam *micro-SD Card*.

Arduino Mega adalah mikrokontroler yang digunakan untuk membaca hasil pengukuran tegangan dari rangkaian pengukuran tegangan dan arus pada PCB *power amplifier* dan juga digunakan untuk membaca tegangan *output* yang dihasilkan tiap detik untuk membuat hasil pembacaan menjadi gelombang serta digunakan untuk menentukan hasil pengecekan *quality control*. Kabel penghubung adalah kabel yang digunakan untuk menghubungkan jarum pengukuran pada rangkaian pengukuran tegangan, modul ACS712, modul sensor LDR, modul *micro-SD Card adapter* dan modul LCD 16x2 I2C. Komputer digunakan untuk menampilkan hasil pengukuran tegangan dan arus dari *Arduino Mega* dan menampilkan gelombang dengan menggunakan software *Better Serial Plotter*. Modul LCD 16x2 I2C adalah layar yang digunakan *Arduino Mega* untuk menampilkan hasil pengecekan *quality control*.

3.1 Perancangan Hardware

Perancangan hardware terbagi menjadi 5 bagian yaitu perancangan rangkaian, perancangan *wiring* modul sensor LDR dan modul ACS712, perancangan *wiring* modul LCD 16x2 I2C, perancangan *wiring* modul *micro-SD Card*, perancangan *Jig* PCB dan desain PCB pengukuran tegangan. Perancangan rangkaian adalah perancangan rangkaian pengukuran tegangan yang akan digunakan. Perancangan *wiring* modul ACS712 dan modul sensor LDR adalah perancangan pemasangan modul ACS712 dan modul sensor LDR pada *Arduino Mega*. Perancangan *wiring* modul LCD 16x2 I2C adalah perancangan pemasangan modul LCD 16x2 I2C pada *Arduino Mega*. Perancangan *wiring* modul *micro-SD Card adapter* adalah perancangan pemasangan modul *micro-SD Card adapter* pada *Arduino Mega*. Perancangan *Jig* PCB adalah melakukan desain *Jig* PCB yang akan digunakan dan merealisasikannya. Pembuatan PCB adalah mendesain dan membuat PCB untuk rangkaian pengukuran tegangan yang akan digunakan.

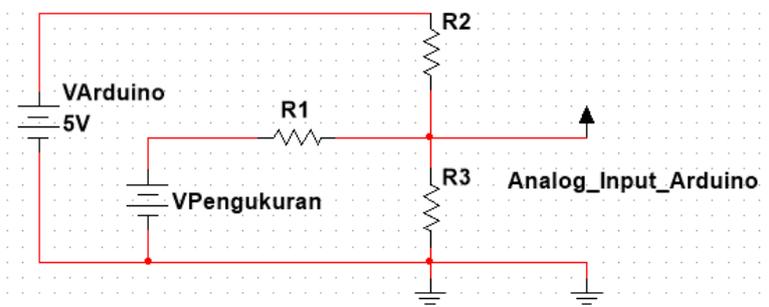
3.1.1 Perancangan Rangkaian



Gambar 3.2 Skema Rangkaian Pengukuran Tegangan

Rangkaian pengukuran tegangan adalah rangkaian pembagi tegangan yang dibuat sesuai dengan skematik rangkaian pada gambar 3.2. Rangkaian ini menggunakan tegangan 5 Volt dari *Arduino Mega* sebagai tegangan referensi. $V_{\text{pengukuran}}$ adalah tegangan pada titik di PCB *power amplifier* yang akan diukur oleh *Arduino Mega*. Bagian positif dan negatif $V_{\text{pengukuran}}$ akan tersambung pada jarum pengukuran pada *Jig PCB*. Rangkaian ini akan digunakan sebanyak 15 rangkaian untuk melakukan pengukuran pada 15 titik dimana 13 rangkaian digunakan untuk mengukur tegangan -15 Volt hingga 15 Volt dan 2 rangkaian digunakan untuk melakukan pengukuran tegangan -30 Volt hingga 30 Volt.

13 rangkaian memiliki tegangan minimum yang akan diukur adalah -15 Volt dan tegangan maksimal yang akan diukur adalah 15 Volt. Sedangkan tegangan yang bisa diukur oleh *Arduino Mega* adalah 0 Volt hingga 5 Volt. Oleh karena itu, ditentukan bahwa tegangan yang bisa diukur oleh *Arduino Mega* adalah 20 Volt dan -20 Volt sehingga perlu menentukan dan menghitung nilai resistor yang akan digunakan. Berikut adalah perhitungan resistor yang akan digunakan ketika tegangan yang akan diukur adalah -20 Volt dengan menggunakan teorema *Thevenin*.



Gambar 3.3 Gambar Rangkaian Pembagi Tegangan Untuk Perhitungan

Mencari R_{th} dengan melakukan *short circuit* pada R3

$$R_3 = R_{load} = 10K\Omega$$

$$R_2 = 10K\Omega$$

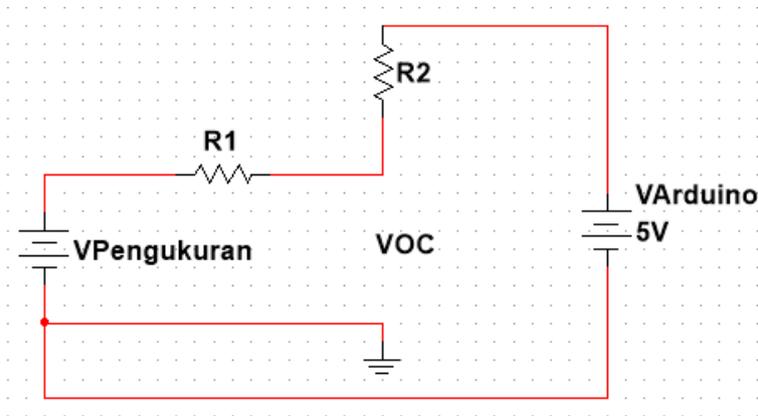
$$R_1 = X$$

$$R_{th} = \frac{1}{\frac{1}{R_2} + \frac{1}{X}}$$

$$R_{th} = \frac{1}{\frac{1}{10K\Omega} + \frac{1}{X}} = \frac{1}{\frac{(X+10K\Omega)}{10K\Omega X}}$$

$$R_{th} = \frac{10K\Omega X}{(X+10K\Omega)} \quad (3.1)$$

Mencari I_{th} dan V_{oc} dengan melakukan *open circuit* pada R3



Gambar 3.4 Gambar Rangkaian Saat Melakukan *Open Circuit* pada R3

$$VPengukuran - 5V = I_{th} \times (10K\Omega + X)$$

$$I_{th} = \frac{VPengukuran - 5V}{10K\Omega + X} \quad (3.2)$$

$$V_{oc} = VPengukuran - I_{th} \cdot X$$

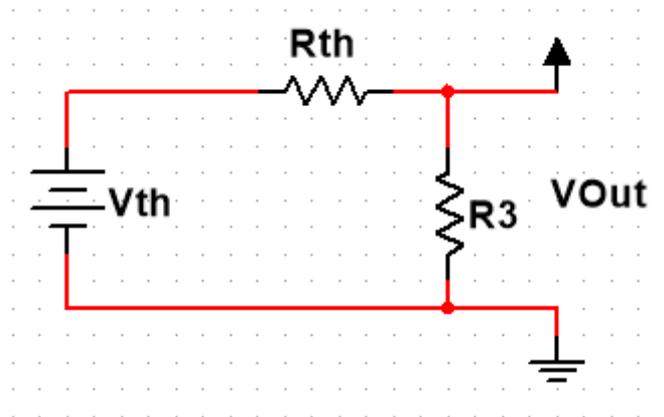
$$V_{oc} = VPengukuran - \frac{VPengukuran - 5V}{10K\Omega + X} \cdot X$$

$$V_{oc} = VPengukuran - \frac{X(VPengukuran - 5V)}{10K\Omega + X}$$

$$V_{oc} = \frac{VPengukuran(10K\Omega + X)}{10K\Omega + X} - \frac{X(VPengukuran - 5V)}{10K\Omega + X}$$

$$V_{oc} = \frac{10K\Omega \cdot VPengukuran + X \cdot 5V}{10K\Omega + X} = V_{th} \quad (3.3)$$

Setelah menemukan V_{oc} dan R_{th} dari perhitungan, rangkaian *thevenin* dibuat dari perhitungan di atas dan mencari X dengan menggunakan rumus yang ditemukan pada rangkaian *thevenin* dimana tegangan output adalah 0 Volt dengan tegangan yang masuk pada rangkaian adalah -20 Volt.



Gambar 3.5 Gambar Rangkaian *Thevenin*

$$V_{Out} = \frac{R_3}{R_{th} + R_3} \times V_{th} \quad (3.4)$$

Substitusi persamaan persamaan 3.1 dan 3.3 ke 3.4

$$V_{Out} = \frac{10K\Omega}{\frac{10K\Omega X}{(X+10K\Omega)} + 10K\Omega} \times \frac{10K\Omega \cdot VPengukuran + X.5V}{10K\Omega + X}$$

$$V_{Out} = \frac{10K\Omega}{\frac{100K^2 + 20K\Omega X}{(X+10K\Omega)}} \times \frac{10K\Omega \cdot VPengukuran + X.5V}{10K\Omega + X}$$

$$V_{Out} = \frac{10K\Omega}{100K^2 + 20K\Omega X} \times (10K\Omega \cdot VPengukuran + X.5V)$$

$$V_{Out} = \frac{10K\Omega \cdot VPengukuran + X.5V}{10K\Omega + 2X} \quad (3.5)$$

Setelah itu, V_{out} disubstitusi dengan angka 0 V dan $V_{pengukuran}$ disubstitusi dengan angka -20 V pada persamaan 3.5 untuk mencari X

$$0V = \frac{10K\Omega \cdot -20V + X.5V}{10K\Omega + 2X}$$

$$0V = -200K\Omega + X.5V$$

$$200K\Omega = X.5V$$

$$40K\Omega = X$$

Setelah menemukan nilai resistor untuk pengukuran tegangan -20 Volt, dilakukan juga perhitungan untuk menemukan $V_{pengukuran}$ pada saat V_{out} memiliki nilai 5 Volt. Perhitungan ini digunakan untuk menentukan batas maksimal tegangan yang bisa diukur oleh rangkaian pembagi tegangan. Batas maksimal tegangan akan digunakan untuk perhitungan mengubah tegangan yang diterima oleh *Arduino Mega* yang akan dijelaskan pada bagian 3.2.1. Perhitungan ini menggunakan persamaan 3.5 untuk menemukan $V_{pengukuran}$.

Substitusi V_{out} dengan angka 5 V dan X dengan $40K\Omega$ pada persamaan 3.5 untuk mencari $V_{pengukuran}$

$$5 V = \frac{10K\Omega \cdot V_{Pengukuran} + 40K\Omega \cdot 5 V}{10K\Omega + 2.40K\Omega}$$

$$5 V = \frac{10K\Omega \cdot V_{Pengukuran} + 40K\Omega \cdot 5 V}{90K\Omega}$$

$$450 VK\Omega = 10K\Omega \cdot V_{pengukuran} + 200VK\Omega$$

$$250 VK\Omega = 10K\Omega \cdot V_{pengukuran}$$

$$25 V = V_{pengukuran}$$

2 rangkaian akan digunakan mengukur tegangan -30 Volt hingga 30 Volt. Karena tegangan minimal yang akan diukur adalah -30 Volt, maka ditentukan tegangan minimal yang bisa diukur oleh rangkaian adalah -40 Volt. Rangkaian yang digunakan sama dengan rangkaian yang digunakan pada perhitungan diatas. Oleh karena itu, perhitungan langsung menggunakan persamaan 3.5 untuk menentukan nilai resistor yang akan digunakan.

Substitusi V_{out} dengan angka 0 V dan $V_{pengukuran}$ dengan angka -40 V pada persamaan 3.5 untuk mencari X

$$0 V = \frac{10K\Omega \cdot -40 V + X \cdot 5 V}{10K\Omega + 2X}$$

$$0 V = -400K\Omega + X \cdot 5 V$$

$$400K\Omega = X \cdot 5 V$$

$$80K\Omega = X$$

Setelah menemukan X, substitusi V_{out} dengan angka 0 V dan X dengan $80K\Omega$ untuk menemukan batas maksimal tegangan yang bisa diukur.

$$5 V = \frac{10K\Omega \cdot V_{Pengukuran} + 80K\Omega \cdot 5 V}{10K\Omega + 2.80K\Omega}$$

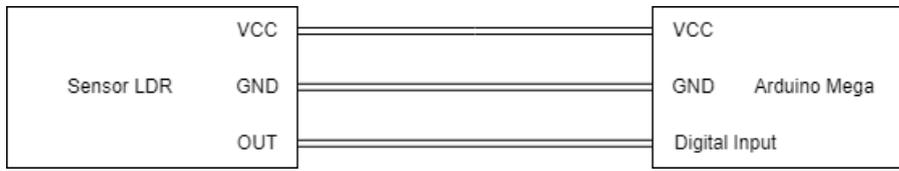
$$5 V = \frac{10K\Omega \cdot V_{Pengukuran} + 80K\Omega \cdot 5 V}{170K\Omega}$$

$$850 VK\Omega = 10K\Omega \cdot V_{pengukuran} + 400VK\Omega$$

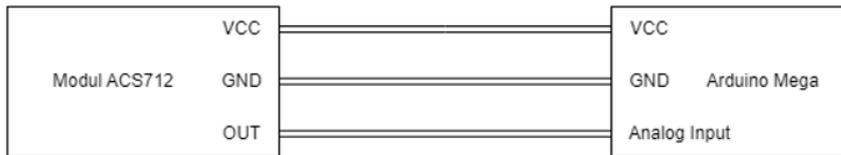
$$450 VK\Omega = 10K\Omega \cdot V_{pengukuran}$$

$$45 V = V_{pengukuran}$$

3.1.2 Perancangan *Wiring* Modul Sensor LDR dan Modul ACS712



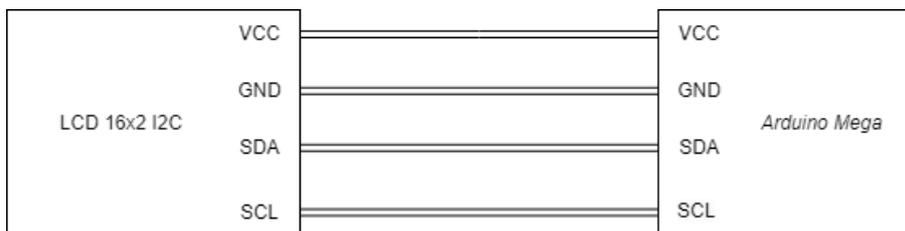
Gambar 3.6 Perancangan Rangkaian Modul Sensor LDR



Gambar 3.7 Perancangan Rangkaian Modul ACS712

Modul sensor LDR adalah modul sensor yang berfungsi untuk mendeteksi PCB power amplifier pada *jig* PCB. Pemasangan modul sensor LDR pada *Arduino Mega* dapat dilihat pada gambar 3.6 dimana *Arduino Mega* akan menerima data dari sensor melalui digital *input*. Modul ACS712 adalah modul yang berfungsi untuk mengukur arus *output* dari PCB *power amplifier*. Pemasangan modul ACS712 pada *Arduino Mega* dapat dilihat pada gambar 3.7 dimana *Arduino Mega* akan menerima data dari sensor melalui analog *input*.

3.1.3 Perancangan *Wiring* Modul LCD 16x2 I2C

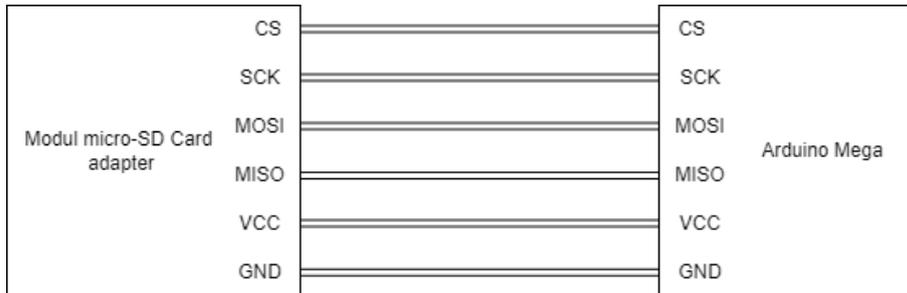


Gambar 3.8 Perancangan *Wiring* Modul LCD 16x2 I2C

Modul LCD 16x2 I2C adalah modul LCD yang digunakan untuk menampilkan tulisan dari *Arduino Mega*. Modul ini digunakan untuk menampilkan status pengecekan *quality control* yang dilakukan oleh *Arduino Mega*. Pemasangan modul LCD 16x2 I2C dapat dilihat pada gambar 3.8 dimana *Arduino Mega* akan mengirimkan data pada modul LCD dengan menggunakan

komunikasi I2C. Pin *Arduino Mega* yang digunakan untuk menggunakan komunikasi tersebut adalah pin SDA pada pin digital 20 dan SCL pada pin digital 21.

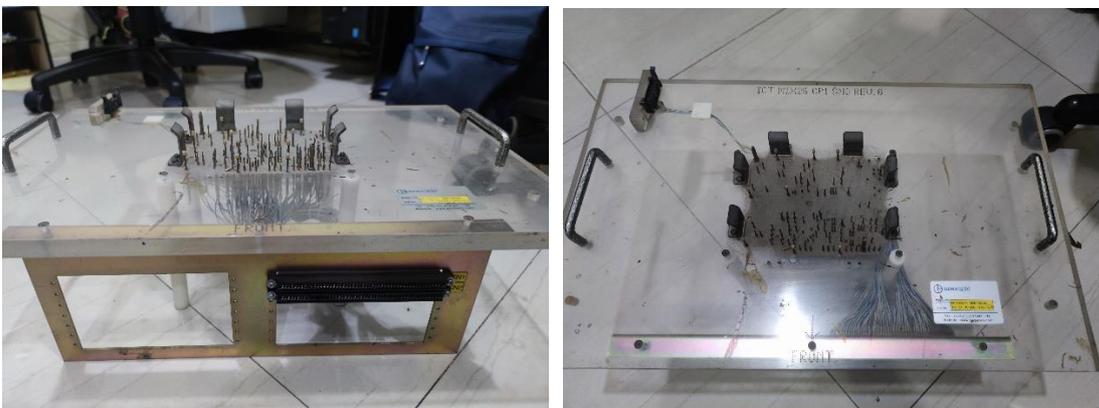
3.1.4 Perancangan *Wiring* Modul *Micro-SD Card Adapter*



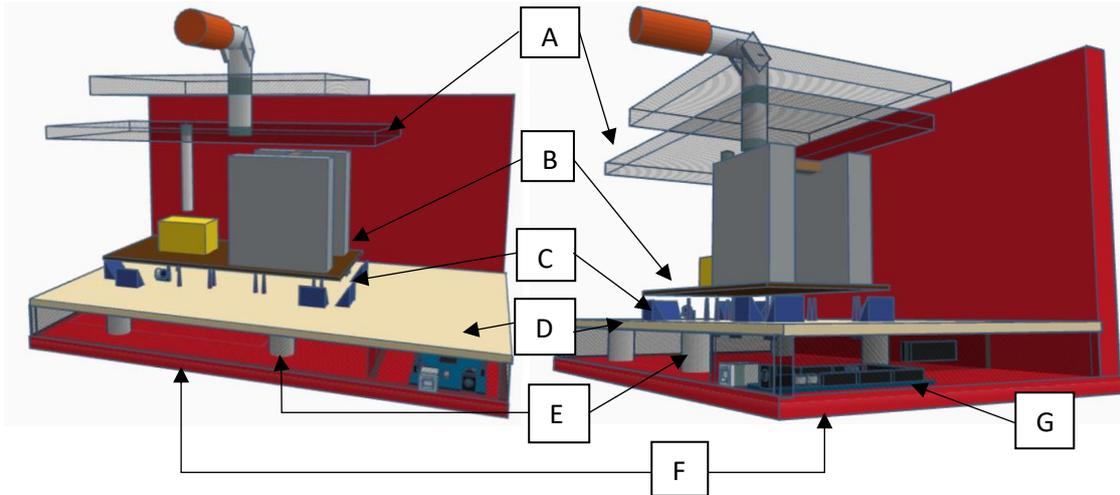
Gambar 3.9 Perancangan *Wiring* Modul *Micro-SD Card Adapter*

Modul *micro-SD Card adapter* adalah modul yang digunakan untuk membaca data dari *micro-SD Card*. Modul ini digunakan untuk membaca *micro-SD Card* yang di dalamnya tersimpan file parameter. Di dalam file parameter terdapat data tegangan maksimal dan tegangan minimal yang akan digunakan untuk proses *quality control*. Pemasangan modul *micro-SD Card adapter* dapat dilihat pada gambar 3.9. Pin yang akan digunakan pada *Arduino Mega* adalah pin digital 53 untuk pin CS, pin digital 52 untuk pin SCK, pin digital 51 untuk pin MOSI dan pin digital 50 untuk pin MISO.

3.1.5 Pembuatan *Jig* PCB



Gambar 3.10 Salah Satu Bentuk *Fixture* ICT PCB



Gambar 3.11 Desain Jig PCB

Keterangan:

A: Penekan PCB *Power Amplifier*

B: PCB *Power Amplifier*

C: Jarum Pengukuran dan Jarum Pengunci

D: Sisi atas *jig* PCB

E: Penyangga Sisi atas *jig* PCB

F: Sisi bawah *jig* PCB

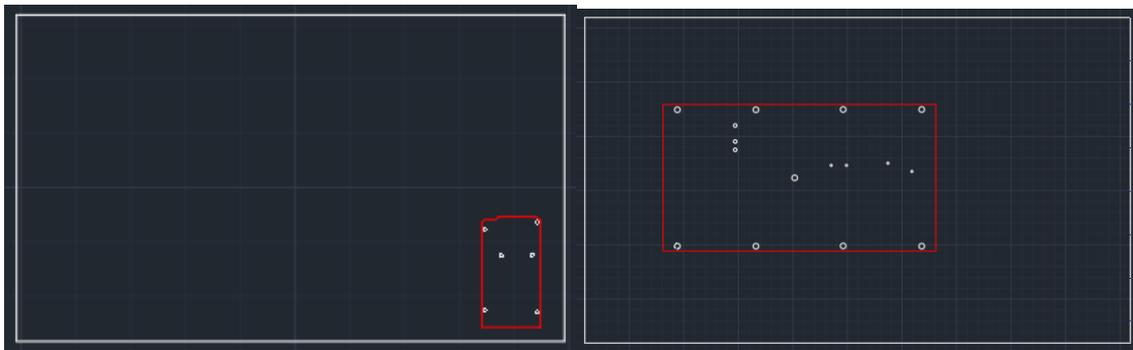
G: *Arduino Mega*

Jig PCB adalah alat bantu yang didesain untuk membuat PCB melayang dari dasar permukaan *jig* PCB sehingga perlu desain rancangan bentuknya sebelum *jig* PCB dibuat. Desain dari *jig* PCB ditunjukkan pada gambar 3.11. *Jig* PCB didesain dengan menggunakan basis dari *fixture* ICT PCB dimana bentuknya ditunjukkan pada gambar 3.10. Desain dari *jig* PCB terdiri dari beberapa bagian dimana bagian tersebut diberi keterangan pada gambar 3.11.

Bagian A adalah bagian yang digunakan untuk menekan PCB *power amplifier*. Bagian B adalah PCB *power amplifier* yang akan dilakukan *quality control*. Bagian C adalah penempatan jarum pengukuran dan jarum pengunci posisi PCB *power amplifier* dimana posisinya disesuaikan dengan beberapa titik di PCB yang perlu dilakukan pengetesan. Bagian D adalah bahan yang digunakan untuk menahan jarum pengukuran dan jarum pengunci. Bagian E adalah bahan yang digunakan untuk menopang bagian D agar terdapat perbedaan tinggi dengan bagian F dimana fungsinya adalah memberikan ruang untuk jarum pengukuran ketika ditekan dan penyusunan kabel dari jarum pengukuran. Bagian F adalah bahan yang digunakan sebagai dasar dari *jig* PCB

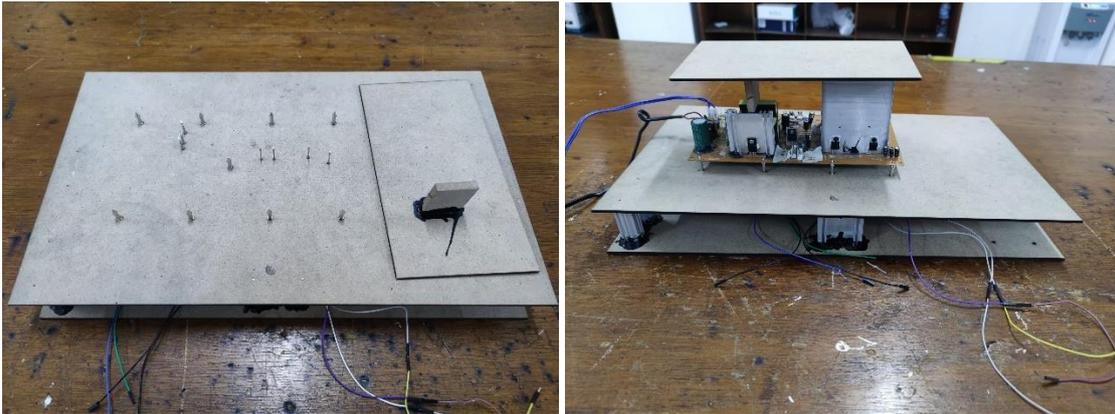
dimana pada bagian ini terpasang modul sensor LDR, modul ACS712 dan modul *micro-SD Card adapter*. Bagian G adalah posisi *Arduino Mega* dalam *jig PCB*.

Selain bagian yang telah disebutkan di atas, terdapat juga sensor LDR dan LCD 16x2 I2C yang akan ditempatkan pada bagian D. Jarum pengukuran akan tersambung dengan *input* pada rangkaian pengukuran tegangan yang akan diletakkan pada sisi kiri posisi *Arduino Mega*. Output dari rangkaian tersebut disambungkan dengan pin analog *Arduino Mega* dimana rangkaian tersebut dijelaskan pada 3.1.6. Kemudian, modul sensor LDR, modul ACS712, modul *micro-SD Card adapter* dan LCD 16x2 I2C juga disambungkan pada *Arduino Mega* sesuai dengan perancangan *wiring* modul sensor LDR dan modul ACS712 yang dijelaskan pada 3.1.2, perancangan *wiring* modul *micro-SD Card adapter* yang dijelaskan pada 3.1.4 dan perancangan *wiring* LCD 16x2 I2C yang dijelaskan pada 3.1.3.



Gambar 3.12 Hasil Desain pada *Software AutoCAD*

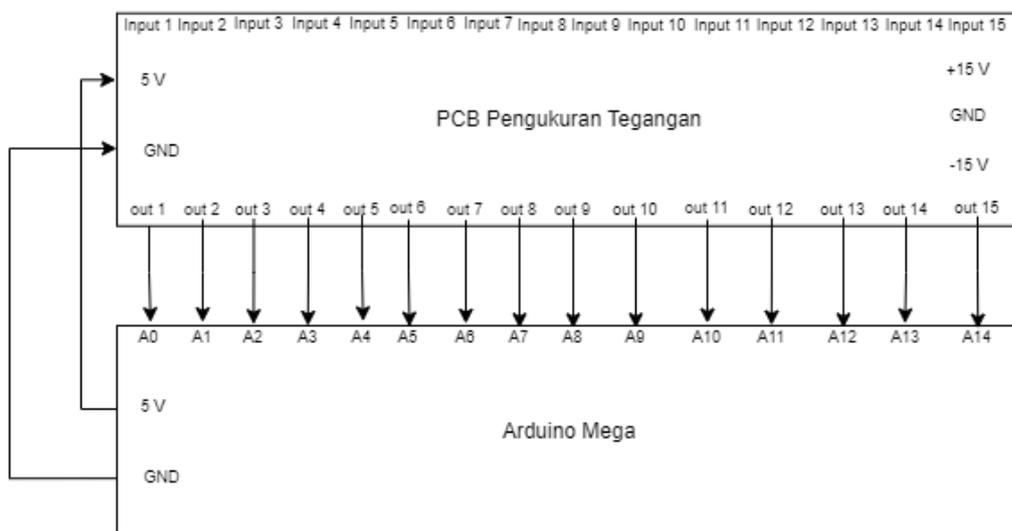
Jig PCB akan dibuat dengan menggunakan bahan kayu MDF sehingga perlu memotong bahan tersebut dengan menggunakan mesin *laser cutting*. Oleh karena itu, *Jig PCB* perlu didesain dengan *software autoCAD* untuk mendesain bentuk secara 2D. Seluruh bagian pada gambar 3.11 perlu didesain dan ditentukan ukurannya. Salah satu desain yang penting adalah perlu mendesain posisi lubang pada bagian D untuk penempatan posisi jarum pengukuran dan jarum pengunci posisi PCB yang ditunjuk oleh bagian C dimana posisi tersebut harus presisi dengan titik pengukuran dari PCB *power amplifier*. Proses dan hasil desain pada *software AutoCAD* yang digunakan untuk *laser cutting* terdapat pada gambar 3.12.



Gambar 3.13 Hasil Jig PCB

Hasil realisasi desain *jig* PCB dapat dilihat pada gambar 3.13. Pada hasil realisasi, terdapat beberapa perbedaan dari desain rancangan. Salah satunya adalah bagian A pada gambar 3.11 yang berfungsi untuk menekan PCB *power amplifier*. Bagian tersebut dibuat terpisah dari *jig* PCB karena membatasi ruang gerak pada saat meletakkan dan mengganti PCB *power amplifier* pada *jig* PCB.

3.1.6 Desain PCB Pengukuran Tegangan

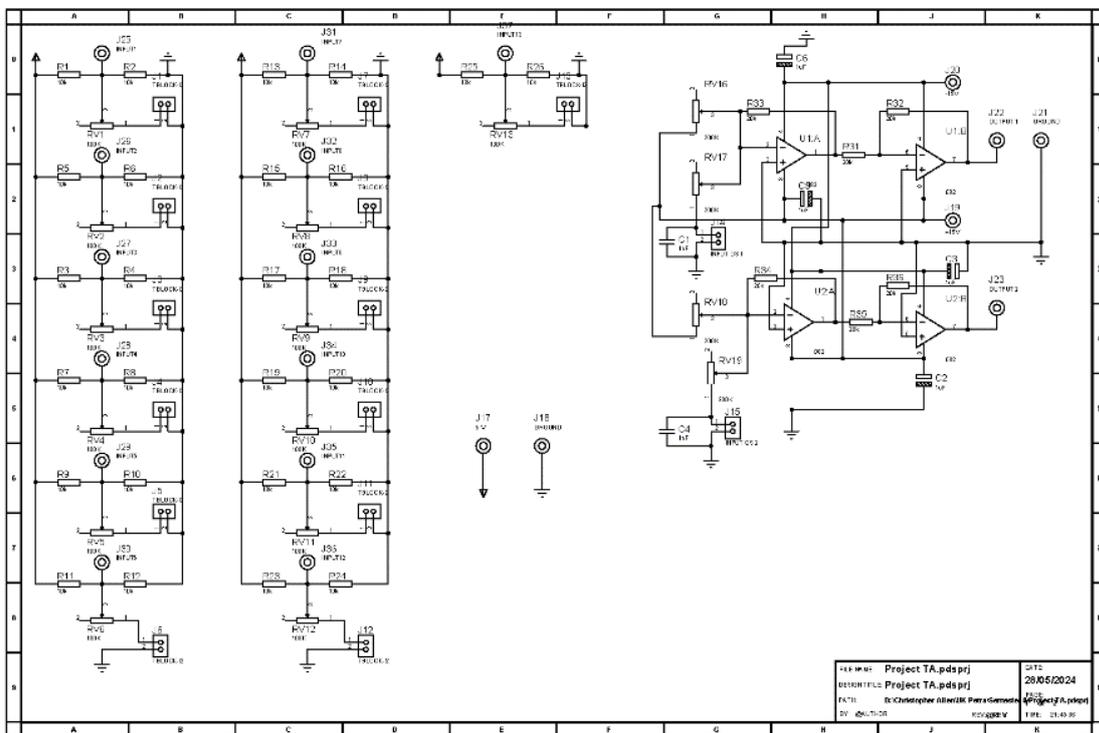


Gambar 3.14 Perancangan PCB Pengukuran Tegangan

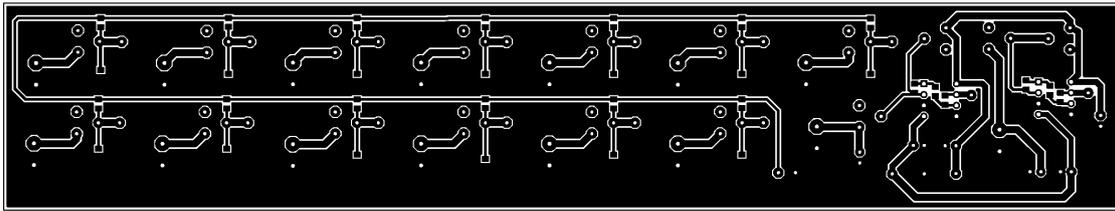
Jig PCB didesain untuk bisa melakukan pengukuran pada 15 titik. Oleh karena itu, 15 rangkaian yang digunakan untuk mengukur tegangan akan digabung menjadi satu PCB untuk mempermudah penempatan pada *jig* PCB. 15 rangkaian yang akan digunakan adalah 13

rangkaian pengukuran tegangan dan 2 rangkaian *zero and span*. Rangkaian pengukuran tegangan digunakan untuk mengukur tegangan pada 13 titik. Kemudian, 2 rangkaian *zero and span* digunakan untuk membaca gelombang dari output PCB *power amplifier*. Rangkaian *zero and span* menggantikan rangkaian pengukuran tegangan yang berfungsi untuk membaca gelombang. Hal ini dikarenakan rangkaian pembagi tegangan tidak cocok digunakan untuk pembacaan gelombang.

Perancangan PCB pengukuran tegangan dapat dilihat pada gambar 3.14. Pada perancangan tersebut, PCB akan memiliki 15 *input* pengukuran dan 15 output pengukuran. Output tersebut akan dihubungkan dengan pin analog *Arduino Mega* dimana pin analog yang digunakan adalah A0 hingga A14. Selain itu, PCB akan menggunakan tegangan dari *Arduino Mega* untuk mensuplai tegangan 5 Volt dan menggunakan *power supply* untuk mensuplai tegangan +15 Volt dan -15 Volt. Tegangan 5 Volt digunakan untuk tegangan referensi rangkaian pengukuran tegangan dimana rangkaian tersebut telah dijelaskan pada 3.1.1. Tegangan +15 Volt dan -15 Volt digunakan untuk menyuplai rangkaian *zero and span* karena rangkaian tersebut menggunakan IC dimana rangkaian *zero and span* akan dijelaskan pada bagian 4.2.



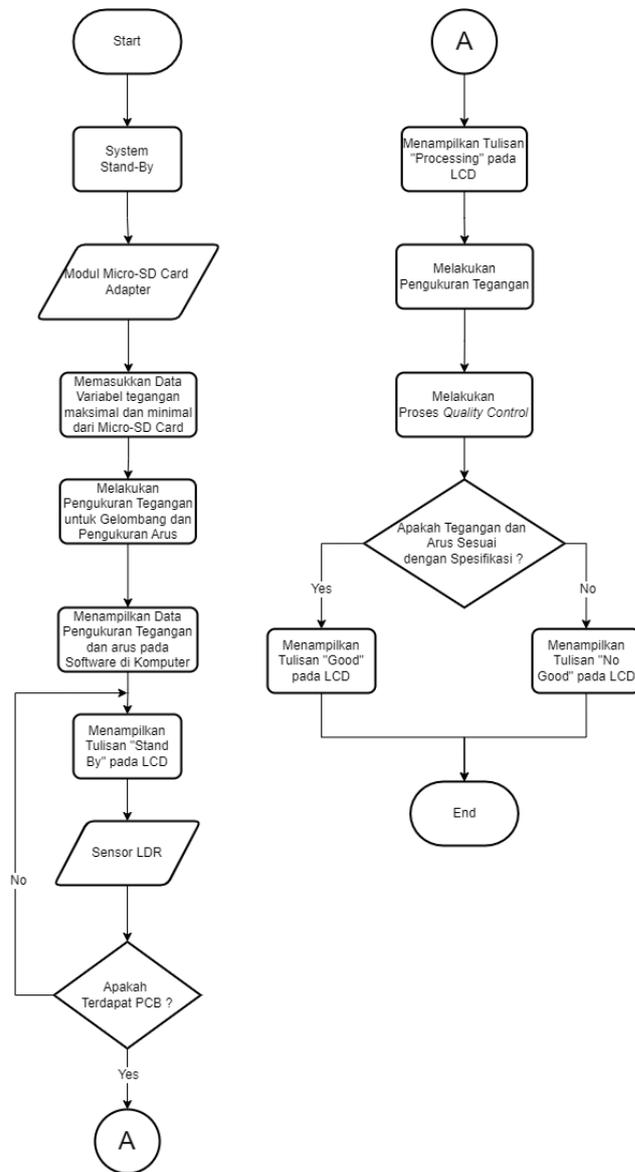
Gambar 3.15 Desain *Schematic* PCB Pengukuran Tegangan



Gambar 3.16 Desain *layout* PCB Pengukuran Tegangan

Proses mendesain PCB terbagi menjadi dua bagian yaitu desain *schematic* PCB dan desain *layout* PCB. Desain *schematic* PCB adalah merealisasikan perancangan PCB yang terdapat pada gambar 3.14. Pada desain *schematic* PCB, terdapat 15 rangkaian yang dibuat dimana gambar desain dapat dilihat pada gambar 3.15. 15 rangkaian tersebut terdiri dari 13 rangkaian pembagi tegangan dan 2 rangkaian *zero and span*. Setelah desain *schematic* telah selesai, dilanjutkan dengan melakukan desain *layout* PCB dimana hasil desain dapat dilihat pada gambar 3.16.

3.2 Perancangan Software



Gambar 3.17 Flowchart Modul Uji Rangkaian Power Amplifier untuk Quality Control

Gambar 3.17 adalah *flowchart* dari modul uji rangkaian *power amplifier* untuk *quality control*. Ketika modul uji ini menyala, *Arduino Mega* membaca data yang tersimpan pada *micro-SD Card* dengan menggunakan modul *micro-SD Card adapter*. Kemudian, data yang tersimpan pada *micro-SD Card* tersebut dimasukkan pada variabel tegangan maksimal dan minimal yang digunakan untuk *quality control*. Setelah itu, *Arduino Mega* melakukan pengukuran tegangan untuk gelombang dan pengukuran arus. Kemudian, semua data pengukuran tegangan dan arus ditampilkan pada *software Better Serial Plotter*. Setelah itu, *Arduino Mega* mendapatkan data

dari sensor LDR yang berfungsi untuk mendeteksi apakah terdapat PCB *power amplifier* yang akan dilakukan pengecekan. Jika tidak mendeteksi, maka akan menampilkan tulisan “*Stand By*” pada LCD.

Jika sensor mendeteksi terdapat PCB, *Arduino Mega* akan menampilkan tulisan “*Processing*” pada LCD dan mulai melakukan pengukuran tegangan dan proses *quality control* pada PCB *power amplifier* tersebut. Setelah itu, *Arduino Mega* menentukan apakah PCB yang sedang dilakukan pengecekan memiliki hasil pengukuran tegangan yang tidak melebihi batas maksimal dan kurang dari batas minimal. Jika hasil pengukuran sesuai dengan spesifikasi atau tidak melebihi dari batas maksimal atau kurang dari batas minimal, maka *Arduino Mega* menampilkan tulisan “*GOOD*” pada LCD. Jika hasil pengukuran melebihi atau kurang dari batas, maka *Arduino Mega* menampilkan tulisan “*NO GOOD*” pada LCD. Oleh karena itu, diperlukan perancangan *software* pada *Arduino Mega* dan koneksi dengan *software better serial plotter*.

Pada saat proses melakukan pengukuran tegangan untuk gelombang dan pengukuran arus, PCB *power amplifier* disambungkan dengan *tone control* yang diberi *input* gelombang sinus dengan frekuensi 1 KHz. Kemudian, *Arduino Mega* mengirim data tegangan pada titik *output* yang diukur perdetik ke komputer yang kemudian akan ditampilkan hasil pengukurannya dalam bentuk gelombang pada *software Better Serial Plotter*. Selain itu, data hasil pengukuran arus juga dikirim untuk ditampilkan hasil pengukurannya pada *software*. Proses ini dilakukan terpisah dari proses *quality control* untuk menghindari efek pembebanan pada PCB *power amplifier* terutama pada bagian proses *input* pada PCB *power amplifier*.

3.2.1 Perancangan Software Arduino Mega

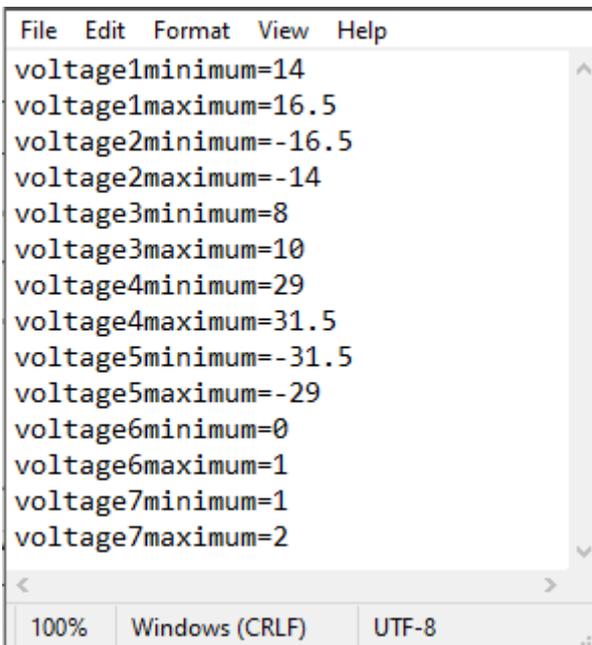
Perancangan alur software sesuai dengan *flowchart* yang telah dibuat dan dapat dilihat pada gambar 3.17. *Software* pada *Arduino Mega* menggunakan 5 *library* yaitu *Wire*, *LiquidCrystal_I2C*, *Filters*, *SPI* dan *SD*. *Library Wire* digunakan pada *Arduino Mega* untuk berkomunikasi dengan LCD 16x2 I2C dimana komunikasi yang digunakan adalah komunikasi I2C (*Inter-Integrated Communication*). *Library LiquidCrystal_I2C* digunakan untuk mengontrol LCD 16x2 I2C yang tersambung dengan komunikasi I2C *Arduino Mega*. *Library Filters* digunakan untuk melakukan perhitungan *input* dari modul ACS712 untuk menjadi arus. *Library SPI* digunakan untuk komunikasi *Arduino Mega* dengan modul *micro-SD Card adapter* karena modul tersebut berkomunikasi dengan menggunakan SPI. *Library SD* digunakan untuk membuat *Arduino Mega* bisa membaca dan menulis data pada *micro-SD Card*. *Library SD* dan SPI sangat terkait karena

dua *library* tersebut berfungsi untuk membaca dan berkomunikasi dengan modul *micro-SD Card adapter*. Gambar 3.18 adalah gambar penggunaan *library* pada software *Arduino Mega*.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Filters.h>
#include <SPI.h>
#include <SD.h>
```

Gambar 3.18 *Library* yang Digunakan pada *Software Arduino Mega*

Pada saat *Arduino Mega* menyala, *Arduino Mega* membaca *file* yang berisi tegangan maksimal dan tegangan minimal yang tersimpan pada *micro-SD Card*. *File* yang digunakan menggunakan format *text* (.txt). Di dalam *file* tersebut terdapat 7 tegangan minimal dan 7 tegangan maksimal karena proses *quality control* hanya mengukur 7 titik. Cara penulisan di dalam *file* terdapat pada gambar 3.19, dimana penulisan nama variabel dan posisi tanda sama dengan tidak dapat diubah karena *Arduino Mega* hanya membaca variabel dengan posisi tersebut. Bagian yang bisa diubah hanya pada angka saja karena angka tersebut dibaca sebagai *value* yang akan dimasukkan pada variabel yang terdapat di dalam *Arduino Mega*.



```
File Edit Format View Help
voltage1minimum=14
voltage1maximum=16.5
voltage2minimum=-16.5
voltage2maximum=-14
voltage3minimum=8
voltage3maximum=10
voltage4minimum=29
voltage4maximum=31.5
voltage5minimum=-31.5
voltage5maximum=-29
voltage6minimum=0
voltage6maximum=1
voltage7minimum=1
voltage7maximum=2
100% Windows (CRLF) UTF-8
```

Gambar 3.19 Cara Penulisan di Dalam *File*

Cara *Arduino Mega* membaca *file* dalam *micro-SD Card* adalah melakukan identifikasi *micro-SD Card* terlebih dahulu. Jika *micro-SD Card* telah teridentifikasi, *Arduino Mega* akan membuka *file* yang tersimpan di dalam *micro-SD Card* sesuai dengan nama *filenya*. Setelah membuka *file*, dilakukan pembacaan 7 tegangan minimal dan 7 tegangan maksimal yang tersimpan di dalamnya. Gambar 3.20 menunjukkan cara *Arduino Mega* membuka dan membaca *file*.

```
void setup() {
  if (!SD.begin(53)) {
    Serial.println("Initialization failed!");
    return;
  }
  Serial.println("Initialization done.");

  File myFile = SD.open("PARAM~1.txt");
  if (myFile) {
    Serial.println("PARAM~1.txt");

    while (myFile.available()) {
      String data = myFile.readStringUntil('\n');

      int separatorIndex = data.indexOf('=');
```

Gambar 3.20 Cara *Arduino Mega* Membuka dan Membaca *File* dalam *Micro-SD Card*

Di dalam *Arduino Mega* juga tersimpan 7 variabel tegangan minimal dan 7 variabel tegangan maksimal. Pada saat pembacaan, *Arduino Mega* akan memasukkan masing-masing nilai tegangan minimal dan tegangan maksimal yang terdapat dalam *file* pada masing-masing variabel tegangan minimal dan maksimal yang terdapat di dalam *Arduino Mega* ketika mendeteksi nama variabel dan tanda sama dengan. Nilai tersebut akan dimasukkan sesuai dengan nama variabelnya. Setelah selesai melakukan pembacaan, *Arduino Mega* akan menutup *file*. Beberapa cara *Arduino Mega* memasukkan nilai variabel dari *file* dapat dilihat pada gambar 3.21.

```

if (separatorIndex != -1) {
    String variableName = data.substring(0, separatorIndex);
    String valueString = data.substring(separatorIndex + 1);
    float value = valueString.toFloat();

    if (variableName == "voltage1minimum") {
        voltage1minimum = value;
    } else if (variableName == "voltage1maximum") {
        voltage1maximum = value;
    } else if (variableName == "voltage2minimum") {
        voltage2minimum = value;
    } else if (variableName == "voltage2maximum") {
        voltage2maximum = value;
    }
}
}
myFile.close();
} else {
    Serial.println("Error opening PARAM~1.txt");
}
}
}

```

Gambar 3.21 Cara *Arduino Mega* memasukkan nilai variabel dari *file* dalam *Micro-SD Card*

Setelah membaca file dalam *micro-SD card*, *Arduino Mega* mendapatkan *input* tegangan dan gelombang dari PCB pengukuran tegangan. Tetapi tegangan yang didapatkan hanya 0 Volt hingga 5 Volt sehingga nilai tersebut perlu dikonversi menjadi tegangan minimal dan maksimal yang telah ditentukan sesuai dengan perancangan rangkaian pada bagian 3.1.1. Cara mengubah nilai tersebut adalah menggunakan rumus persamaan garis lurus dimana rumus ini diubah menjadi bahasa agar dapat melakukan perhitungan pada *Arduino Mega*. Rumus persamaan garis lurus dapat dilihat pada persamaan 3.6. Gambar 3.22 adalah rumus persamaan garis lurus yang diubah menjadi bahasa untuk digunakan pada *software Arduino Mega*.

$$\begin{aligned}
 y - y_1 &= \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) \\
 y &= \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) + y_1
 \end{aligned}
 \tag{3.6}$$

```
float mapFloat(float x, float in_min, float in_max, float out_min, float
out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

Gambar 3.22 Rumus persamaan Garis Lurus pada *Software Arduino Mega*

Setelah memasukkan rumus persamaan garis lurus pada *software*, rumus itu digunakan untuk mengubah nilai yang masuk pada pin analog *Arduino Mega*. Sebelum memasukkan nilai *input* pada rumus, nilai tersebut perlu diubah menjadi nilai tegangan terlebih dahulu. Hal ini dikarenakan nilai *input* dibaca oleh *Arduino Mega* menggunakan *analog to digital converter* dengan resolusi 10-bit yaitu dari angka 0 -1023. Cara mengubah nilai *input* menjadi tegangan adalah nilai *input* dikali dengan 5 dan dibagi dengan resolusi 10-bit yaitu 1024 dimana cara ini dijelaskan pada persamaan 3.7. Gambar 3.23 adalah cara mengubah nilai *input* dari membaca nilai *input* dari pin analog, kemudian mengubah nilai *input* menjadi nilai tegangan dan memasukkan nilai tegangan pada rumus persamaan garis lurus. Nilai 5 Volt dalam program *Arduino Mega* diubah menjadi 4,75 Volt dimana penjelasannya dapat dilihat pada bagian 4.4.

$$\text{tegangan} = \frac{\text{nilai input}}{5V} \cdot 1024 \quad (3.7)$$

```
float data2 = analogRead(A3);
float adc_value2 = data2 * 4.75 / 1024;

voltage2 = mapFloat(adc_value2, 0.0, 4.75, -20, 25);
```

Gambar 3.23 Salah Satu Proses Konversi Nilai *Input* pada *Software Arduino Mega*

Setelah membuat proses konversi nilai *input* dan tegangan pada *software*, dilanjutkan dengan mengukur nilai arus. Nilai tersebut didapatkan dari modul ACS712 tetapi masih dalam bentuk nilai resolusi 10-bit. Selain itu, arus yang diukur adalah arus AC (*Alternate Current*) sehingga tidak bisa mendapatkan nilai arus secara langsung. Oleh karena itu, *library filters* digunakan untuk membaca nilai dari modul ACS712 dan mencari nilai arus RMS dengan menggunakan fungsi statistik dari *library*. Gambar 3.24 menjelaskan cara menggunakan *library* dan memasukkan nilai dari modul ACS712 pada fungsi statistik *library*.

```

float ACS_Value;
float windowLength = 0.1;
RunningStatistics currentStats;
float intercept = 0;
float slope = 0.0752;
float Amps_TRMS;
unsigned long sampleRate = 2000;
unsigned long sampleInterval = 1000000 / sampleRate;
unsigned long lastSampleTime = 0;
unsigned long printPeriod = 10;
unsigned long previousMillis = 0;

void setup() {
  Serial.begin(115200);
  pinMode (A15, INPUT);
  currentStats.setWindowSecs(windowLength);
}

void loop() {
  if (waktu - lastSampleTime >= sampleInterval) {
    lastSampleTime += sampleInterval;
    float currentRawValue = analogRead(A15);
    currentStats.input(currentRawValue);
    if ((unsigned long)(millis() - previousMillis) >= printPeriod) {
      previousMillis = millis();
      Amps_TRMS = intercept + slope * currentStats.sigma();
    }
  }
}

```

Gambar 3.24 Pengukuran Arus dari Modul ACS712 dengan menggunakan *Library Filters*

Setelah membuat pengukuran arus, dilanjutkan dengan membuat pengukuran gelombang. Proses pengukuran arus dan pengukuran gelombang berjalan dengan bersamaan dan tidak bergantung dengan sensor LDR. Proses pengukuran gelombang memiliki cara yang sama dengan melakukan pengukuran tegangan. Salah satu proses melakukan pengukuran tegangan dapat dilihat pada gambar 3.25.

```

void signal_reading(){
    float data = analogRead(A0);
    float adc_value = data * 4.75 / 1024;

    signal1 = mapFloat(adc_value, 0.0, 4.75, -20, 20);
}

```

Gambar 3.25 Salah Satu Proses Melakukan Pengukuran Tegangan

Setelah membuat pengukuran arus, dilanjutkan dengan membuat sistem pendeteksi PCB pada *jig* PCB menggunakan sensor LDR. Berdasarkan dari *flowchart*, jika sensor LDR mendeteksi terdapat PCB, maka akan menampilkan tulisan “PROCESSING...” pada LCD 16x2 I2C dan melakukan pengukuran tegangan. Dengan *input* sensor LDR pada *Arduino Mega* adalah digital *input*, maka tidak ada perhitungan yang dilakukan. Selain itu, sensor LDR juga berfungsi sebagai kondisi untuk menampilkan tulisan pada LCD 16x2 I2C dan juga berfungsi untuk memulai proses *quality control*. Salah satu proses melakukan pengukuran tegangan dapat dilihat pada gambar 3.26. Cara menampilkan tulisan “PROCESSING...” pada LCD 16x2 I2C dapat dilihat pada gambar 3.27.

```

if (distance == 1){

    MeasureVoltage();

void MeasureVoltage(){

    float data1 = analogRead(A2);
    float adc_value1 = data1 * 4.75 / 1024;

    voltage1 = mapFloat(adc_value1, 0.0, 4.75, -20, 25);
    //Titik 1
}
}

```

Gambar 3.26 Salah Satu Proses Melakukan Pengukuran Tegangan

```

if (!displayed){
    screen.clear();
    screen.setCursor(0,0);
    screen.print("PROCESSING...");
    delay (2000);
    displayed = true;
    screen.clear();
    delay (100);
}

```

Gambar 3.27 Cara Menampilkan tulisan "PROCESSING..." pada LCD 16x2 I2C

Proses *quality control* adalah menentukan apakah PCB yang sedang diukur dalam keadaan "GOOD" atau "NO GOOD". Proses ini dilakukan dengan menggunakan batas tegangan minimal dan batas tegangan maksimal yang didapatkan dari file pada *micro-SD Card*. Jika hasil pengukuran tegangan tidak melebihi batas pengukuran, maka akan menampilkan tulisan "GOOD" pada LCD 16x2 I2C. Proses melakukan *quality control* dapat dilihat pada gambar 3.28.

```

else {
    if (voltage1 > voltage1minimum && voltage1 < voltage1maximum){
        if (voltage2 > voltage2minimum && voltage2 < voltage2maximum){
            if (voltage3 > voltage3minimum && voltage3 < voltage3maximum){
                if (voltage4 > voltage4minimum && voltage4 < voltage4maximum){
                    if (voltage5 > voltage5minimum && voltage5 < voltage5maximum){
                        if (voltage6 > voltage6minimum && voltage6 <
voltage6maximum){
                            if (voltage7 > voltage7minimum && voltage7 <
voltage7maximum){
                                screen.setCursor(0,0);
                                screen.print("GOOD");
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Gambar 3.28 Proses Melakukan *Quality Control*

Jika hasil pengukuran tegangan melebihi batas tegangan maksimal atau kurang dari batas tegangan minimal, maka akan menambahkan tulisan "NO GOOD" pada LCD 16x2 I2C. Selain itu, ditampilkan juga tulisan bagian yang tegangannya melebihi atau kurang dari batas pengukuran. Tulisan itu diletakkan di bawah tulisan "NO GOOD" dan bergerak dari kanan ke kiri. Gambar 3.29 menunjukkan proses ketika hasil pengukuran tegangan lebih atau kurang dari batas pengukuran dan gambar 3.30 menunjukkan *function* untuk membuat tulisan bergerak dari kanan ke kiri.

```
if (voltage1 < voltage1minimum || voltage1 > voltage1maximum) {
    errormessages += "TITIK 15 V ";
}
if (voltage2 < voltage2minimum || voltage2 > voltage2maximum) {
    errormessages += "TITIK -15 V ";
}
if (voltage3 < voltage3minimum || voltage3 > voltage3maximum) {
    errormessages += "TITIK 9 V ";
}
if (voltage4 < voltage4minimum || voltage4 > voltage4maximum) {
    errormessages += "TITIK 30 V ";
}
if (voltage5 < voltage5minimum || voltage5 > voltage5maximum) {
    errormessages += "TITIK -30 V ";
}
if (voltage6 < voltage6minimum || voltage6 > voltage6maximum) {
    errormessages += "TITIK 0.6 V ";
}
if (voltage7 < voltage7minimum || voltage7 > voltage7maximum) {
    errormessages += "TITIK 1.2 V ";
}
if (errormessages != "") {
    screen.clear();
    screen.setCursor(0,0);
    screen.print("NO GOOD");
    screen.setCursor(0, 1);
    scrollMessage(1, errormessages, 200, 16);
}
}
```

Gambar 3.29 Proses Ketika Hasil Pengukuran Tegangan Lebih atau Kurang dari Batas Pengukuran

```

void scrollMessage(int row, String message, int delayTime, int totalColumns)
{
  for (int i=0; i < totalColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int position = 0; position < message.length(); position++) {
    screen.setCursor(0, row);
    screen.print(message.substring(position, position + totalColumns));
    delay(delayTime);
  }
}

```

Gambar 3.30 *Function* untuk Membuat Tulisan Bergerak dari Kanan ke Kiri Pada LCD 16x2 I2C

Jika, sensor LDR tidak mendeteksi PCB *Power Amplifier*, maka akan menampilkan tulisan “STAND BY” pada LCD 16x2 I2C. Selain itu, semua variabel pengukuran yang terdapat pada *Arduino Mega* dikosongkan. Hal ini untuk mencegah terjadinya variabel pengukuran yang masih menyimpan nilai pengukuran sebelumnya. Proses tersebut dapat dilihat pada gambar 3.31.

```

else{
  screen.setCursor(0,0);
  screen.print("STAND BY");
  displayed = false;
  voltage1 = 0;
  voltage2 = 0;
  voltage3 = 0;
  voltage4 = 0;
  voltage5 = 0;
  voltage6 = 0;
  voltage7 = 0;
  errormessages = "";
}

```

Gambar 3.31 Proses Ketika Sensor LDR Tidak Mendeteksi PCB *Power Amplifier*

3.2.2 Better Serial Plotter



Gambar 3.32 Software Better Serial Plotter

Sumber: *Betterserialplotter* (2021). Nathandunk.
<https://github.com/nathandunk/BetterSerialPlotter.git>

Better Serial Plotter adalah pengganti drop in untuk *arduino serial plotter* (Nathandunk, 2021). *Software* ini merupakan *software open source* yang bisa didapatkan dari *website github* dan dibuat oleh nathandunk. *Software* ini berfungsi untuk membaca *serial monitor* yang diberi oleh *Arduino Mega*. Gambar dari *software* ditunjukkan pada gambar 3.32. *Software* ini dapat menerima banyak data angka yang dikirim oleh *Arduino Mega*. Selain itu, *software* ini dapat membentuk data yang diterima dari *Arduino Mega* menjadi gelombang sehingga bisa menampilkan gelombang sinus.

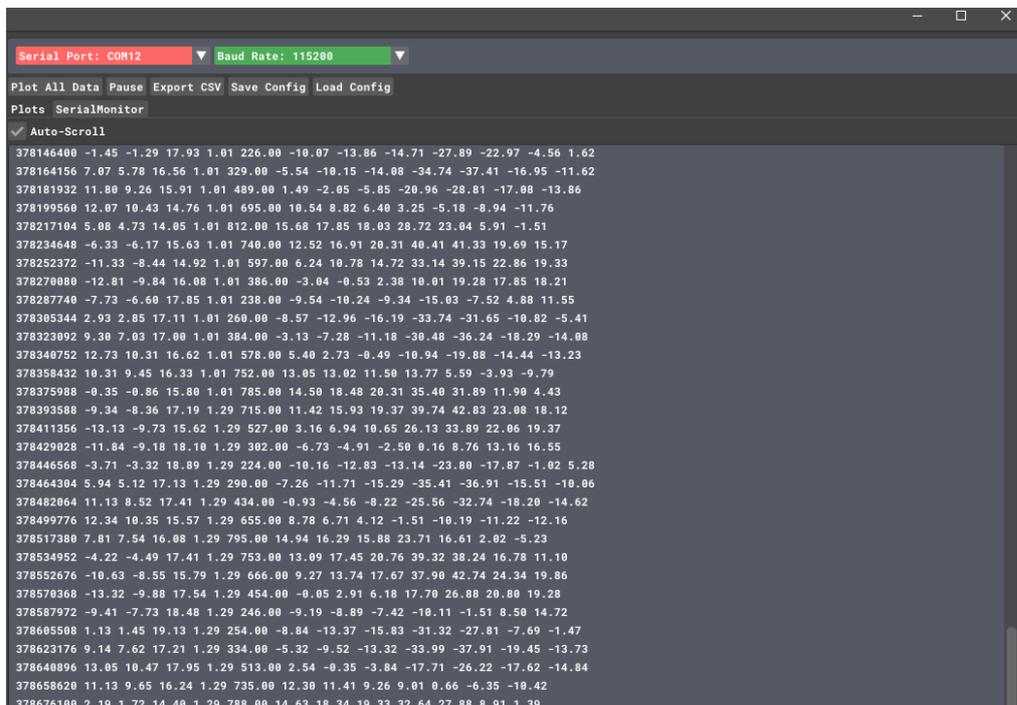
Cara kerja *software* ini adalah *software* menerima sejumlah data dari *Arduino Mega* melalui komunikasi *serial* dari kabel USB (*Universal Serial Bus*). *Arduino Mega* mengirimkan data melalui komunikasi serial dengan menggunakan fungsi *Serial.print*. Setiap data yang dikirim pada *software* diberi spasi antar data untuk membuat *software* dapat membaca dan membedakan data dengan menggunakan fungsi *Serial.print(" ")*. Setelah semua data yang dikirim telah selesai, pada data terakhir menggunakan fungsi *Serial.println (data)* untuk berpindah baris ke bawah (*Cartriged Return*). Hal ini dilakukan untuk memberikan data terbaru lagi. Cara melakukan pengiriman data dari *Arduino Mega* pada *software better serial plotter* dapat dilihat pada gambar 3.33 dan hasil pembacaan pada *software better serial plotter* dapat dilihat pada gambar 3.34.

```

void printvalue(){
    Serial.print (signal1);
    Serial.print (" ");
    Serial.print (signal2);
    Serial.print (" ");
    Serial.print (Amps_TRMS);
    Serial.print (" ");
    Serial.print (voltage1);
    Serial.print (" ");
    Serial.print (voltage2);
    Serial.print (" ");
    Serial.print (voltage3);
    Serial.print (" ");
    Serial.print (voltage4);
    Serial.print (" ");
    Serial.print (voltage5);
    Serial.print (" ");
    Serial.print (voltage6);
    Serial.print (" ");
    Serial.println (voltage7);
}

```

Gambar 3.33 Cara Pengiriman Data pada Software Better Serial Plotter



Gambar 3.34 Salah Satu Hasil Pembacaan Software Better Serial Plotter