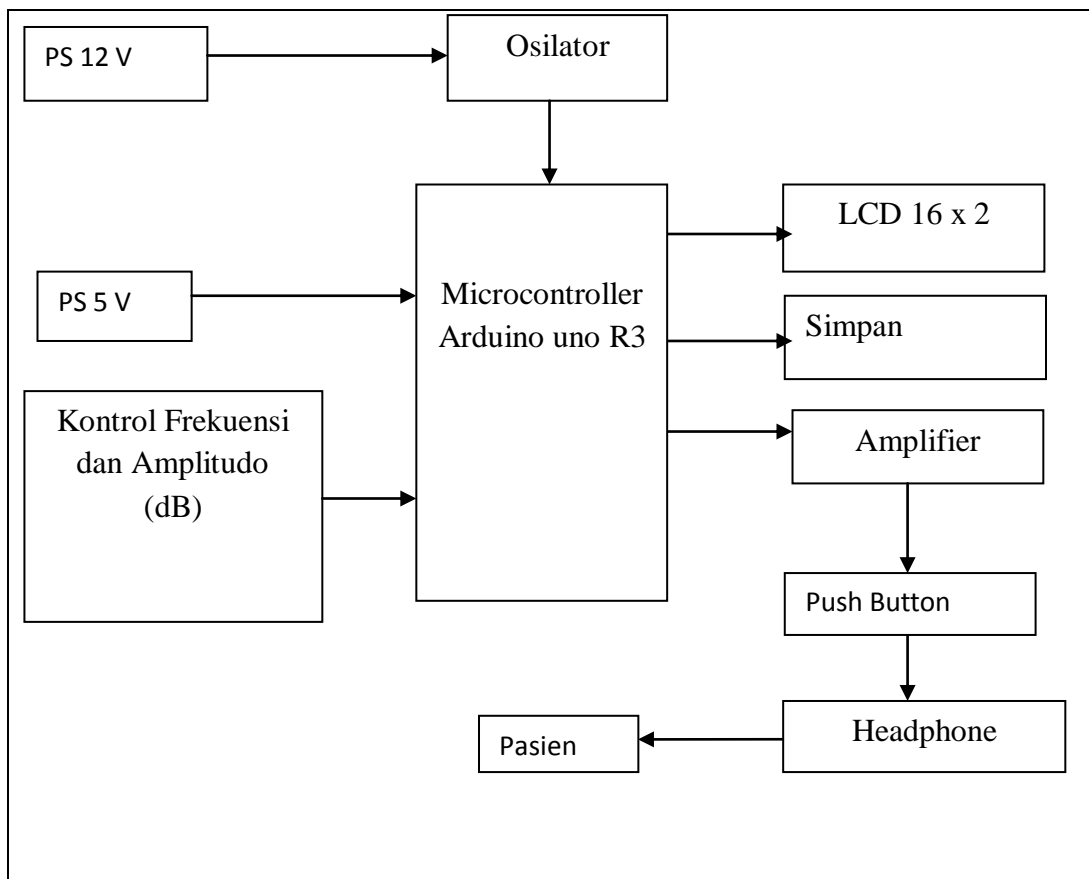


BAB III METODE PENELITIAN

3.1 Diagram Blok Sistem

Blok diagram dapat mempermudah dalam menganalisis masalah pada alat audiometer berbasis arduino uno R3. Blok diagram alat ditunjukkan pada gambar 3.1 berikut ini.



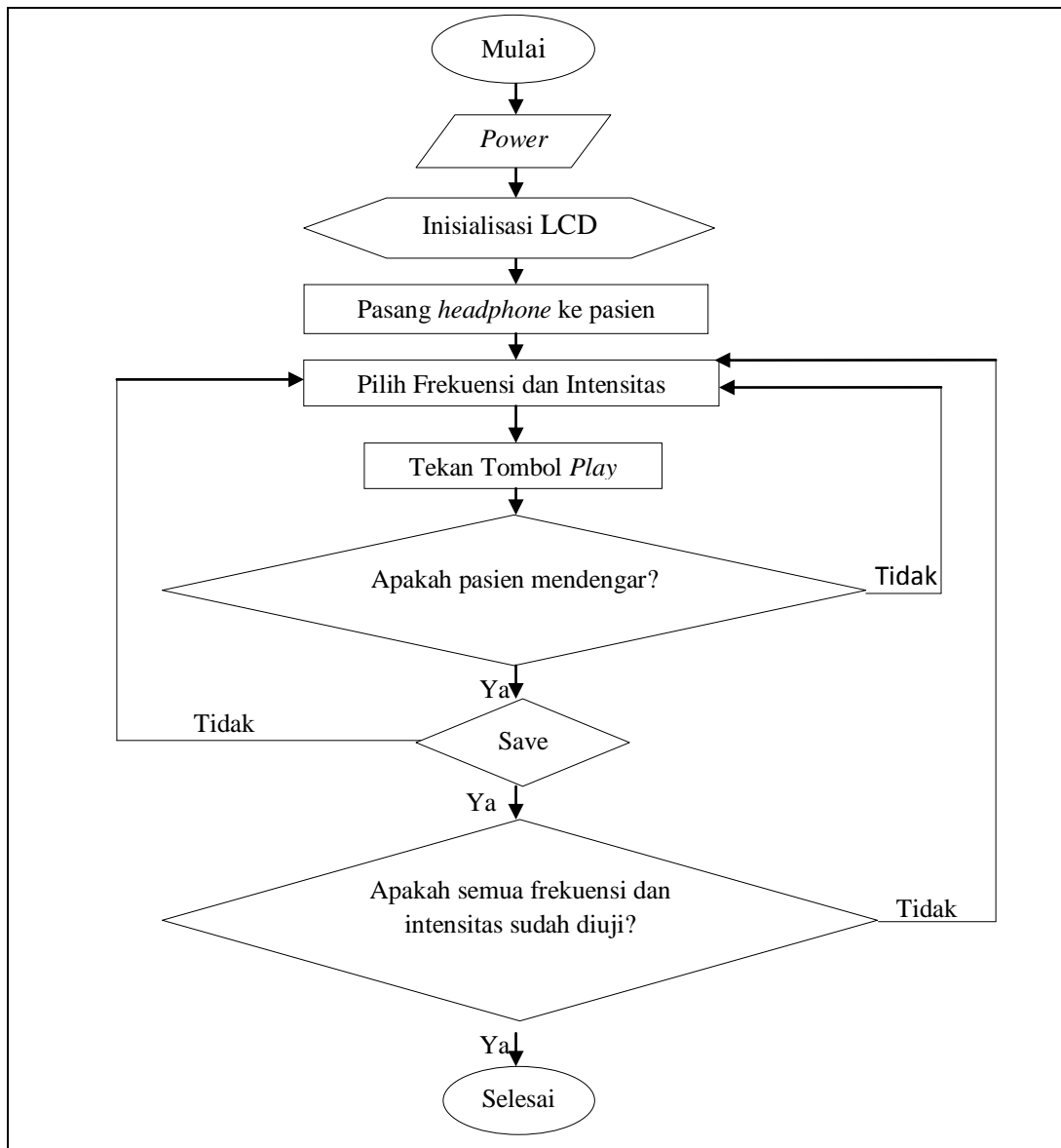
Gambar 3. 1 Blok diagram audiometer berbasis arduino uno R3

Ketika tombol *on* ditekan maka *supply* tegangan akan masuk ke seluruh rangkaian pada alat. Osilator akan membangkitkan frekuensi dengan keluaran gelombang sinus dan kotak. Gelombang sinus digunakan untuk mengatur intensitas bunyi (dB) sedangkan gelombang kotak untuk menghitung nilai

frekuensi. Nilai frekuensi diatur dengan cara memutar resistor *variable* (potensio). Frekuensi akan dihitung oleh *microcontroller* dengan menghitung jumlah *counter* per detiknya, kemudian ditampilkan di *display*. Gelombang sinus akan dikuatkan oleh *amplifier* kemudian dihubungkan dengan *headphone*. Potensio digunakan untuk mengatur intensitas bunyi yang keluar. Bunyi yang telah keluar berupa nada murni yang akan di teruskan ke *headphone*. Supaya bunyi nada tesnya tidak keluar terus menerus, maka ditambahkan saklar supaya bunyi keluar sesuai dengan lama waktu yang dibutuhkan. Jika pasien mendengar maka tombol *save* akan di pencetnya, maka dB dan frekuensi beserta tanggal tes yang ditertampil di *display* akan tersimpan pada *SD card*. Nilai frekuensi dan desibel yang telah diujikan akan disimpan pada *micro sd* secara berurutan berdasarkan nilai frekuensi dan desibel yang diujikan.

3.2 Diagram Alir Proses/Program

Diagram alir adalah urutan cara kerja alat dari awal sampai akhir. Berikut adalah diagram alir audiometer berbasis arduino uno R3 dapat dilihat pada gambar 3.2 .



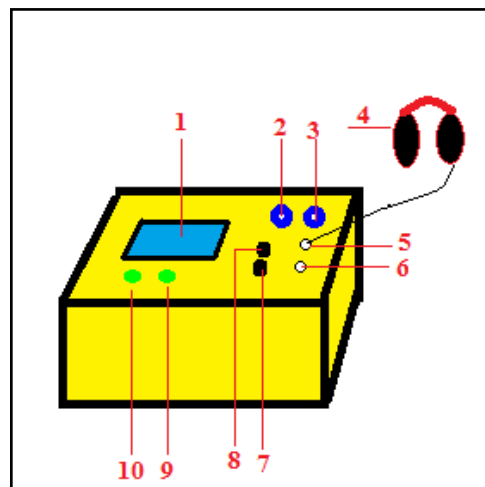
Gambar 3. 2 Diagram alir audiometer berbasis arduino uno R3

Pada saat alat dihidupkan maka akan terjadi inisialisasi LCD, tegangan mengaktifkan osilator. Pasang *headphone* ke pasien. Selanjutnya atur frekuensi dan dB sesuai dengan petunjuk pengetesan pendengaran sesuai standar kesehatan. Tekan tombol *play* untuk mengeluarkan bunyi. Pada saat alat mulai melakukan pengujian lihat reaksi pasien, jika mendengar pasien akan mengangkat tangan, jika pasien telah mendengar maka tekan tombol *save*, lakukan kembali pemilihan

frekuensi dan intensitas suara untuk melakukan pengujian pada frekuensi dan intensitas suara yang lain. Proses akan terus berulang sampai semua frekuensi dan intensitas suara telah diuji. Data pengetesan frekuensi dan desibel yang dapat didengar oleh pasien akan tersimpan pada *micro sd* dengan data simpan yang berurutan.

3.3 Diagram Mekanis Sistem

Mekanis yang penulis buat menggunakan bahan dari akrilik yang sebelumnya sudah didesain sedemikian rupa menggunakan aplikasi *corel draw* sehingga didapatkan hasil bentuk fisik dari *box* audiometer. Berikut adalah bagian-bagian darimekanis alat audiometer.



Gambar 3. 3 Diagram mekanis audiometer berbasis arduino uno R3

Keterangan dari gambar 3.3 yaitu:

1. LCD 16x2, untuk menampilkan nilai frekuensi dan dB.
2. Potensio pengatur frekuensi, untuk memilih frekuensi.
3. Potensio pengatur intensitas (dB), untuk memilih dB.
4. *Headphone*, sebagai tempat keluarnya bunyi.

5. *Jack Output Balance*, untuk menyambungkan ke *headset* dengan mode *balance* (kiri dan kanan).
6. *Jack Output L/R*, untuk menyambung ke *headset* dengan mode L/R.
7. Saklar L/R, untuk memilih telinga yang akan diuji, kiri atau kanan
8. Saklah baterai/adaptor, untuk memilih sumber tegangan, baterai atau adaptor.
9. Tombol *play*, sebagai tombol untuk mengeluarkan bunyi.
10. Tombol *save*, sebagai tombol untuk menyimpan data.

3.4 Alat dan Komponen

1.4.1 Alat

Berikut ini adalah alat-alat yang digunakan dalam pembuatan alat. Seperti yang tertera pada tabel 3.1 dibawah ini :

Tabel 3. 1 Alat yang digunakan

No	Nama	Jumlah
1	Laptop	1
2	<i>Toolset</i>	1
3	Multimeter	1
4	Bor	3
5	Gergaji besi	1
6	<i>Spidol Permanent</i>	2
7	Wadah	1
8	Setrika	1

1.4.2 Komponen

Berikut ini adalah komponen yang digunakan dalam pembuatan alat. Seperti yang tertera pada tabel 3.2 dibawah ini :

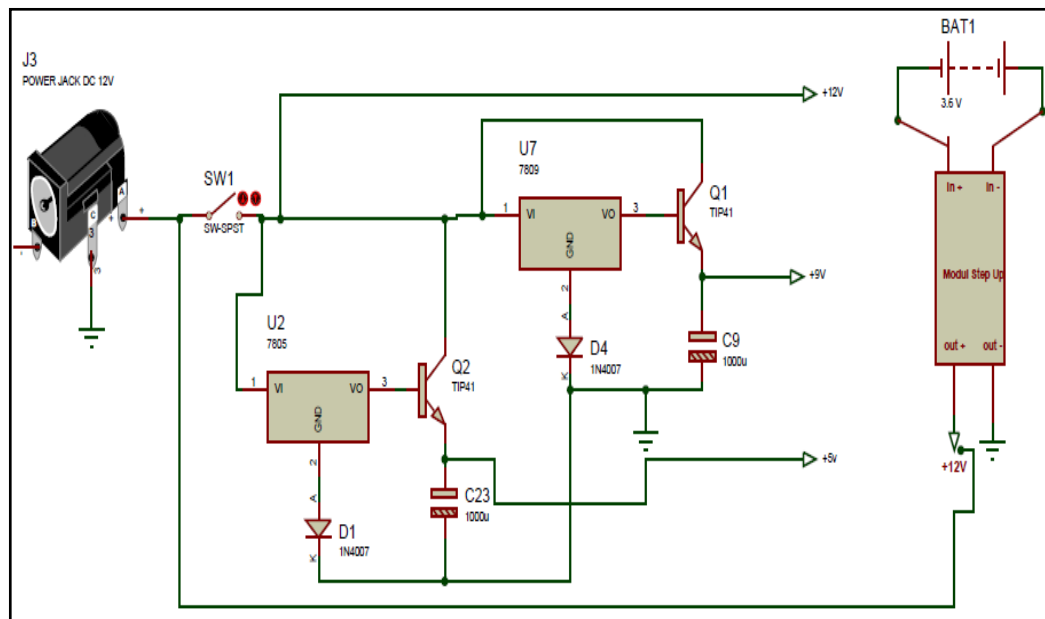
Tabel 3. 2 Komponen yang digunakan

No	Nama Komponen	Jumlah
1	Headphone	1
2	LCD 16 x 2	1
3	Adaptor DC 12 V	1
4	Baterai AA 3 x 1,2 V 1200 mAh	3
5	IC XR 2260	1
6	IC Regulator 7809	1
7	IC Regulator 7805	1
8	Arduino Uno R3 (ATMega328P)	1
9	TL 072	2
10	TC 4584	1
11	TIP 41	2
12	Transistor NPN	4
13	Kristal 16MHz	1
14	Kapasitor 22pF	2
15	Kapasitor 1uF	3
16	Kapasitor 2,2uF	1
17	Kapasitor 10uF	3
18	Kapasitor 100uF	5
19	Kapasitor 470uF	3
20	RV 5K Ohm	1
21	RV 200 Ohm	1
22	RV 50K Ohm	2
23	RV 20K Ohm	1
24	RV 100K Ohm	1
25	Resistor 10K Ohm	21
26	Resistor 100K Ohm	3
27	Resistor 4,7K Ohm	3
28	Resistor 27K Ohm	3
29	Resistor 2,2K Ohm	2
30	Resistor 3,3K Ohm	1
31	Resistor 1K Ohm	2
32	Resistor 1,2K Ohm	2
33	IN4007	1
34	1N4148	1
35	Push Button	2
36	Jack Audio 3,5 mm	2
37	Saklar 2 kaki	1
38	Saklar 3 kaki	2
39	HCL dan H2O2	Secukupnya
No	Nama Komponen	Jumlah
40	Kabel	Secukupnya
41	Printed Circuit Board (PCB)	Secukupnya
42	Modul Charger	1
43	Modul Step Up	1
44	Modul Step Up	1

3.5 Rangkaian Modul

1.5.1 Rangkaian *Power Supply*

Rangkaian *power supply* adalah rangkaian yang berfungsi sebagai pemasok tenaga dan arus listrik untuk rangkaian. Gambar 3.4 menunjukkan rangkaian *power supply*.

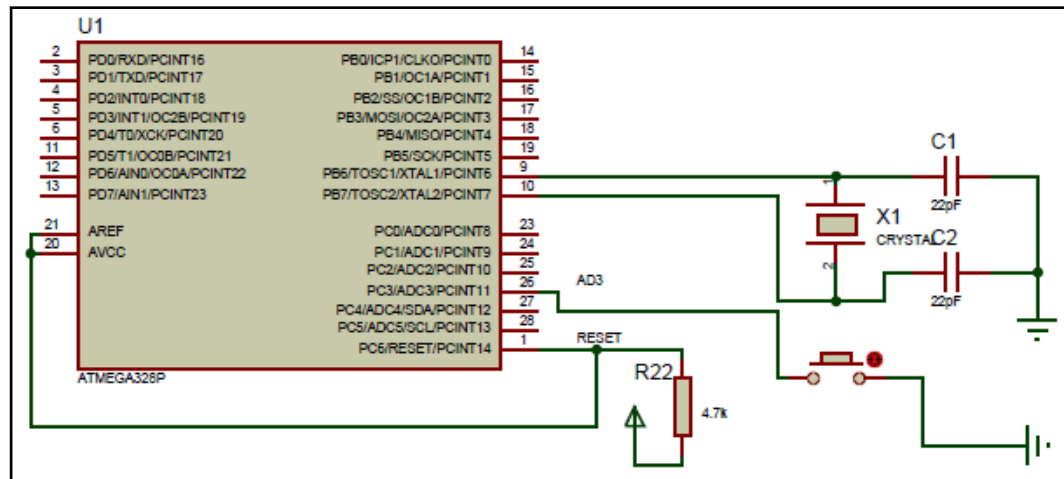


Gambar 3. 4 Rangkaian *Power Supply*

Tegangan 12 V DC bersumber dari *adaptor* dan baterai. Tegangan dari *adaptor* masuk ke saklar. Tegangan baterai adalah 3,6 V, kemudian dinaikkan tegangannya menjadi 12 V dengan menggunakan modul *step up*, kemudian masuk ke saklar. Tegangan 12 V masuk ke *IC regulator 7809* untuk membatasi tegangan keluaran sebesar 9 V, transistor digunakan sebagai penstabil tegangan dan kapasitor digunakan untuk menyaring *ripple* tegangan yang masih bocor. Tegangan 12 V masuk ke *IC regulator 7805* untuk membatasi tegangan keluaran sebesar 5 V, transistor digunakan sebagai penstabil tegangan dan kapasitor digunakan untuk menyaring *ripple* tegangan yang masih bocor.

1.5.2 Rangkaian *Minimum System*

Rangkaian *minimum system* berfungsi sebagai kontrol dari semua modul yang terdapat pada alat. Rangkaian *minimum system* dapat dilihat pada gambar 3.5 dibawah ini.



Gambar 3. 5 Rangkaian *Minimum System*

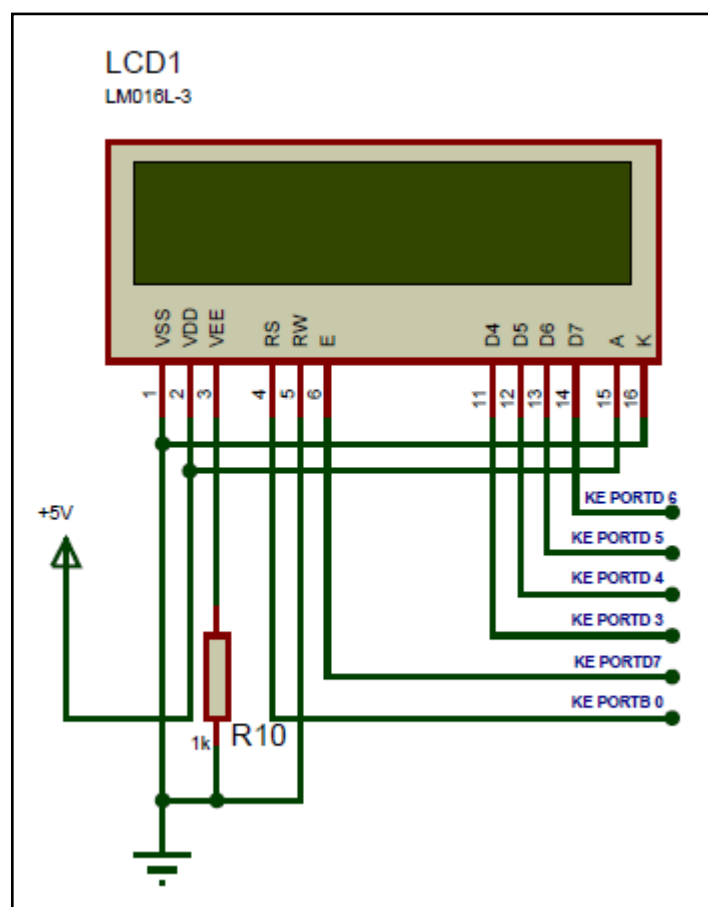
Rangkaian minimum sistem pada modul ini berfungsi sebagai kontrol kerja modul secara keseluruhan. Fungsi dari kristal adalah sebagai *clock* tambahan yang terhubung dengan kapasitor berguna sebagai pengosongan dan pengisian osilator, sebenarnya ATmega328p sudah mempunyai *clock* internal tetapi frekuensinya masih kecil sehingga tidak maksimal dalam pengolahan data. Pada IC ATmega 328p ini diberi program yang akan mengontrol sistem kerja modul. Adapun program yang digunakan pada modul ini menggunakan program arduino.

Berikut ini fungsi *port-port* pada ATmega328p, PD2 tersambung ke IC 4584. PD3 sampai PD7 ditambah PB0 tersambung ke rangkaian LCD. PB2 sampai PB5 terhubung ke modul SD *card*. PB6 dan PB7 masing-masing tersambung ke kaki kristal. PC1 sebagai *input* ADC2, PC2 sebagai *input* ADC1.

PC3 terhubung ke *push button* tombol *save*. PC4 tersambung ke SDA, PC5 tersambung ke SCL pada modul RTC 1307.

1.5.3 Rangkaian LCD

Rangkaian LCD berfungsi untuk menampilkan karakter dan data pengukuran pada alat. Gambar 3.6 dibawah merupakan rangkaian LCD karakter 16 x 2.



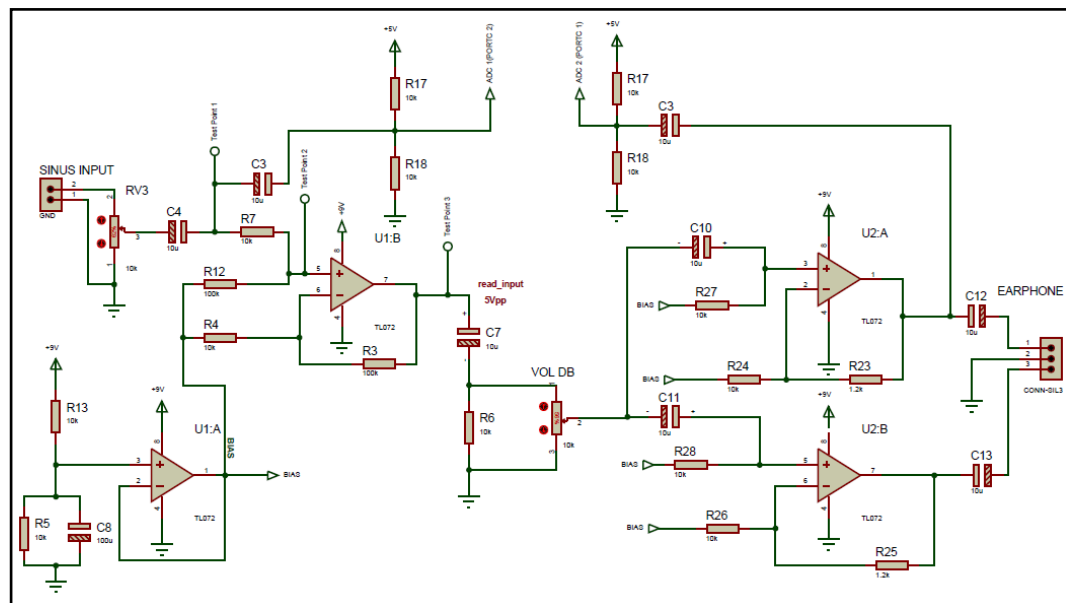
Gambar 3. 6 Rangkaian LCD

LCD yang digunakan adalah 16 x 2, tegangan masukan 5V. pin-pin pada LCD terhubung ke rangkaian *minimum system* sesuai dengan jalur pada gambar.

PIN 2 sebagai keluaran gelombang sinus, kemudian masuk ke rangkaian transistor penguat tegangan. Transistor PNP berfungsi untuk memberikan bias positif pada transistor NPN. Kapasitor 1000 uF berfungsi sebagai kopel antara rangkaian penguat. PIN 11 sebagai keluaran gelombang kotak, kemudian masuk ke rangkaian transistor *buffer* sebagai penyetabil sinyal. Gelombang kotak masuk ke IC 4584 sebagai penegas gelombang kotak kemudian masuk ke mikro untuk nilai menghitung frekuensi.

1.5.5 Rangkaian Amplifier

Rangkaian *amplifier* berfungsi untuk menaikkan tegangan yang masuk ke rangkaian. Gambar 3. 8 adalah gambar rangkaian *amplifier*.



Gambar 3. 8 Rangkaian Amplifier

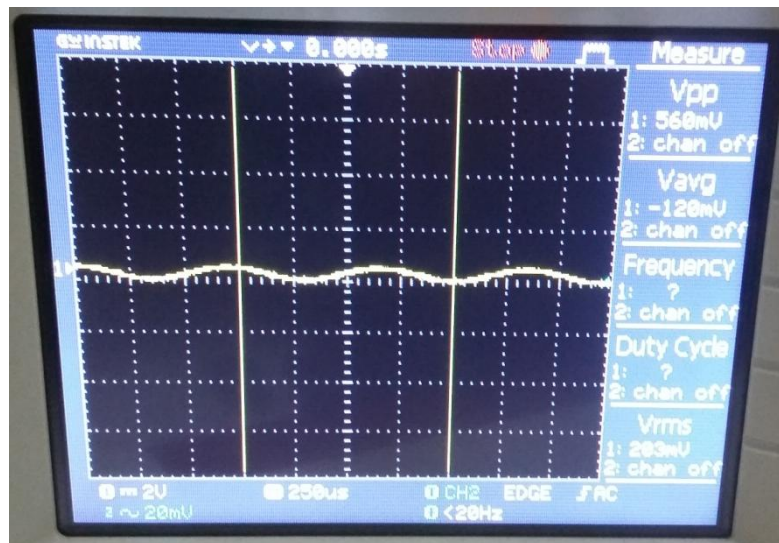
Gelombang sinus, masuk ke potensio (RV3). Potensio berfungsi sebagai pengatur tinggi gelombang sinus yang masuk ke kaki 3 pada U1.B. Setelah potensio dipasang kapasitor C4 yang berfungsi sebagai kopling, fungsinya supaya menahan tegangan DC dari rangkaian selanjutnya supaya tidak saling

mempengaruhi (*Test Point 1*). Setelah dari kapasitor sinyal masuk ke rangkaian yang terhubung ke *PORTC 2* sebagai ADC 2, dengan tegangan referensinya 2,5 V. Sebelum masuk ke kaki 3 U1.B sinyal masuk ke pembagi tegangan pada R7 dan R12 (*Test Point 2*). Potensio (RV3) diatur hingga keluaran *OpAmp U1.B* 5Vpp (*Test Point 3*). Tegangan 5 Vpp merupakan hasil dari penguatan *non-inverting* dengan penguatan 11 kali dengan nilai R3 100K Ohm dan R4 10K Ohm.. U1.A adalah rangkaian *buffer* menguatkan tegangan sebesar 1 kali. Keluarannya di jadikan rangkaian bias atau tegangan referensi supaya *output* sinyal tidak ada yang dibawah garis nol. Keluaran U1.B kemudian masuk ke C7, kemudian masuk ke potencio Vol dB. Potensio Vol dB ini berfungsi untuk mengatur keluaran dari U2.A dan U2.B yang tersusun secara paralel. Penguatan pada U2.A dan U2.B adalah sebesar 1,12 kali dengan nilai R23=R25 yakni 1,2K Ohm dan R24=R28 yakni 10K Ohm. Keluran U2.A digunakan sebagai *input* ADC 1, sedangkan keluaran U2.B disambungkan ke *headphone* . Fungsi kapasitor C13 adalah untuk memblok gelombang DC sehingga yang masuk ke *headphone* hanya gelombang sinus saja.

Adapun rumus untuk menghitung nilai dB berdasarkan tegangan adalah:

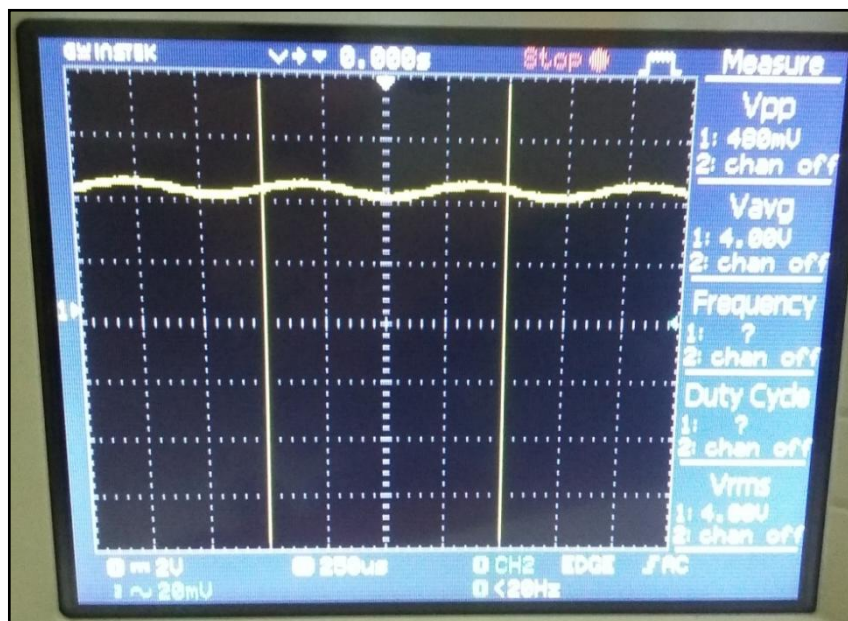
$$\boxed{\text{dB} = 20 \log V_{\text{out}}/V_{\text{in}}}$$
(3-2)

Untuk mempermudah dalam penelitian, maka kita harus mengetahui bentuk gelombang pada masing-masing *test point*. Berikut ini adalah bentuk gelombang pada masing-masing *test point*.



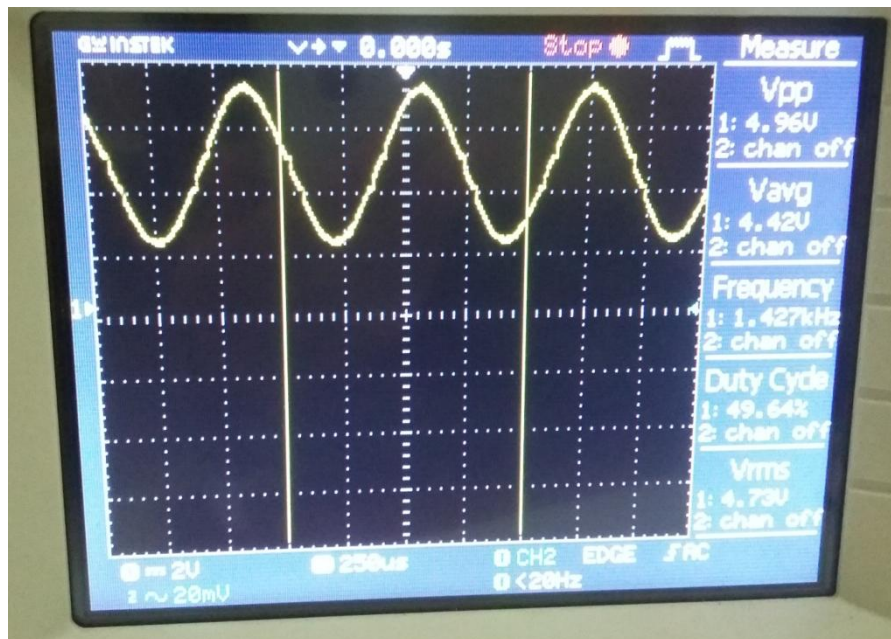
Gambar 3. 9 Bentuk gelombang pada *test point 1*

Sinyal keluaran potensio (RV3) dengan amplitudo 560 mV (Vpp), sebelum masuk ke rangkaian pembagi tegangan.



Gambar 3. 10 Bentuk gelombang pada *test point 2*

Sinyal setelah masuk ke pembagi tegangan sebagai *input* positif, dengan titik referensinya 4,5 Volt.



Gambar 3. 11 Bentuk gelombang pada *test point 3*

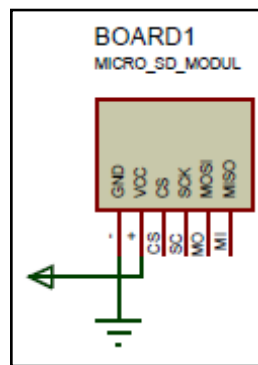
Sinyal keluaran dari U1.B, dengan amplitudo lebih kurang 5 Vpp. Nilai ini adalah hasil dari penguatan dari sinyal test point 2. Berikut adalah perhitungan dari perhitungan dari penguatan keluaran pada U1.B yaitu menggunakan *rumus non-inverting*.

$$\begin{aligned}
 V_{out} &= 1 + (R_f/R_{in}) V_{in} \\
 &= 1 + (100/10) \cdot 0,48 \\
 &= 11 \times 0,48 \\
 &= 5,28 \text{ Vpp}
 \end{aligned}$$

Berdasarkan perhitungan diatas, seharusnya tegangan keluaran pada U1.B jika 11 kali penguatan adalah 5,28 Vpp. Nyatanya tegangan keluaran hanya 4.96 Vpp, ini dikarenakan toleransi dari nilai resistor yang digunakan, sehingga penguatannya tidak pas 11 kali penguatan.

1.5.6 Modul SD card

Modul *SD card* berfungsi sebagai modul penyimpanan data. Berikut adalah gambar rangkaian *SD card* pada gambar 3. 9.

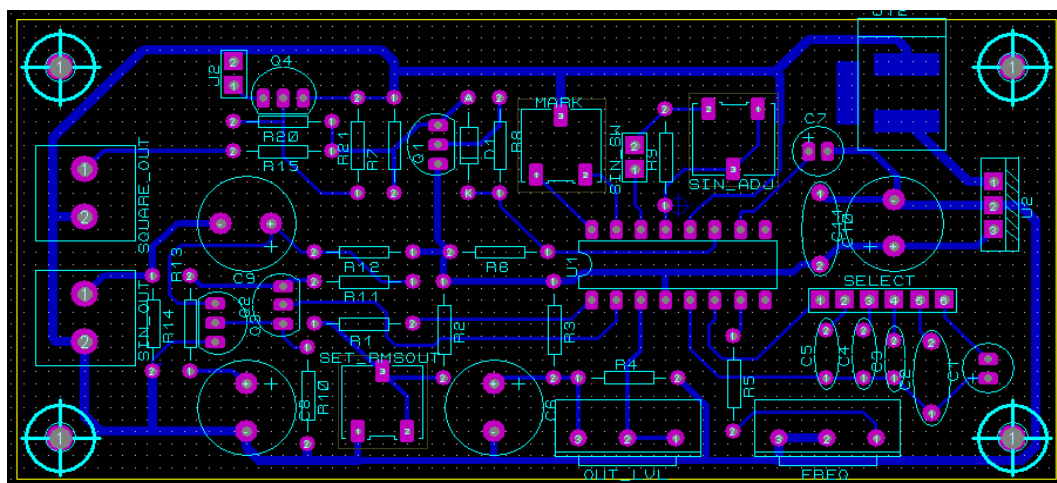


Gambar 3. 12 Modul SD Card

Pin-pin pada *SD card* terhubung dengan *port* pada *minimum* sistem, hubungkan sesuai dengan tulisan yang terdapat pada *SD card*.

3.6 Pembuatan Lay Out

Pembuatan *lay lut* bertujuan untuk membuat jalur skema rangkaian pada papan PCB. Dimana nantinya akan menjadi tempat meletakkannya komponen-komponen yang digunakan. Dengan menggunakan aplikasi Proteus 7.10 pada bagian Ares. Berikut ini adalah *lay out* yang dibuat.



Gambar 3. 13 Lay Out Pembangkit Frekuensi



Gambar 3. 18 Lay Out Amplifier

3.7 Listing Program

3.7.1 Listing program library

```
#include <math.h>
#include <TimerOne.h>
#include <SD.h>
#include <SPI.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 RTC;
File myFile;
int pinCS = 10; // Pin 10 on Arduino Uno
LiquidCrystal lcd(8, 7, 3, 4, 5, 6);
//const int ADC1 = A1, ADC2 = A2;
#define FIRST_ADC_INPUT 1
#define LAST_ADC_INPUT 2
unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];
const int save = A3;
int buttonState = LOW;
int lastButtonState = LOW;
int lastDebounceTime = 0;
int debounceDelay = 50, dat_filt=0;
```

Penjelasan, “`#include <math.h>`” adalah untuk memanggil *library math*, yaitu *library* matematika. Kemudian “`#include <TimerOne.h>`” adalah untuk memanggil *library timerone*. Kemudian “`#include <SD.h>`” berfungsi untuk memanggil *library SD card*. “`#include <SPI.h>`” berfungsi untuk memanggil

library SPI yang berfungsi untuk komunikasi mikro dengan *SD card*. Kemudian “`#include <LiquidCrystal.h>`” berfungsi untuk memanggil *library* LCD. “`#include <Wire.h>`, `#include "RTClib.h"` adalah *library* untuk RTC. “`RTC_DS1307 RTC;`” Tipe RTC yang digunakan. “`File myFile;`” adalah deklarasi file pada *SD card*. Kemudian “`int pinCS = 10;`” adalah memfungsikan Pin 10 arduino sebagai pin CS *SD card*. Konfigurasi LCD “`LiquidCrystal lcd(8, 7, 3, 4, 5, 6);`” Dibawah ini adalah pendefinisian variable-variabel, “`#define FIRST_ADC_INPUT 1, #define LAST_ADC_INPUT 2` pendefinisian ADC 1 dan ADC 2. `unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];` dimensi *array*. “`const int save = A3;`” artinya tombol *save* pada Pin A3. “`int buttonState = LOW; int lastButtonState = LOW; int lastDebounceTime = 0; int debounceDelay = 50,` variable tombol *save*. “`dat_filt=0;`” pengambilan data puncak. “`unsigned int ADC1_temp=0,ADC2_temp=0,ADC2_hsl=0,ADC1_hsl=0;` `#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))`” variabel untuk proses ADC *interrupt*.

Variabel “`int last_adc1=0, last_adc2=0, ADC1 = 0, ADC2 = 0; int lop1=0,lop2=0,msec=0,sec=0,out=0,inpt=0; long period=0,tot_period=0; float dat_des=0,dat_temp=0,hasil,hasil_filter=0; const byte interpt_Pin0 = 2;`” variabel data untuk mengolah ADC. “`String dataString = "";`” merubah tipe data ke data *string*. “`int thn = 0, bln = 0, hri = 0, jam = 0, mnt = 0, dtk = 0;`” variabel data tanggal.

3.7.2 Listing Program RTC

```

Wire.begin();
RTC.begin();
RTC.adjust(DateTime(thn,bln,hri,jam,mnt,dtk));
RTC.adjust(DateTime(2018,8,8,17,13,0));
DateTime now = RTC.now();
//int thn=0,bln=0,hri=0,jam=0,mnt=0,dtk=0;
//thn = now.year(); bln = now.month(); hri = now.day(); jam = now.hour(); mnt =
now.minute(); dtk = now.second();
}

```

Penjelasan, inisialisasi *wire library* kemudian inisialisasi RTC, masuk ke pengaturan tanggal dan waktu pada “*RTC.adjust(DateTime(thn,bln,hri,jam,mnt,dtk));*”. Waktu sekarang sama dengan *RTC now*.

3.7.3 Listing Program Perhitungan db dan Frekuensi

```

void loop() {
  lcd.setCursor(0,0);
  inpt = (inpt + ADC2_hsl)/2;out = (out + ADC1_hsl)/2;
  hasil = (float)out/(inpt);
  dat_des = 20 * log(hasil);
  hasil_filter = (6.8356*dat_des) + 34.99;
  hasil_filter = hasil_filter - 16;
  lcd.print("Intens:");lcd.print((int)hasil_filter); lcd.print(" dB "); lcd.setCursor(0,1);
  lcd.print("Freq:");lcd.print(tot_period);lcd.print(" Hz ");
  int set_mode = digitalRead(save);
}

```

Penjelasan, “*void loop*” artinya program akan dibaca secara berulang. Selanjutnya, “*lcd.setCursor (0,0);*” pada LCD koordinat (0,0) akan tertampil hasil dari perhitungan ADC. *Output* ADC 1 dibagi dengan *input* ADC 2, kemudian masuk rumus dB, yaitu $20 \log(v_{out}/v_{in})$. Karena hasilnya dari minus maka ditambahkan rumus baru seperti pada diatas agar nilainya sesuai dengan *sound*

level. Pada LCD akan tertampil "*Intens*" di baris depan, ditengah tampilan nilai hitung ADC, dibelakang adalah satuan yakni "dB".

Pada LCD koordinat (0,1) akan tertampil frekuensi. Tampilan dibaris depan yakni "*Freq*", tengah adalah hasil perhitungan frekuensi, belakang adalah satuan nyaitu "Hz".