

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

SISTEM *IMPRESSED CURRENT CATHODIC PROTECTION* (ICCP)

ANODA KORBAN PADA PIPA BAJA

(Studi kasus: PT. PLN (Persero) PEMBANGKITAN TANJUNG JATI B

Unit 3&4) BERBASIS ARDUINO UNO

Proses pembuatan benda kerja alat Tugas Akhir yang berjudul “Perancangan Sistem *Cathodic Protection Impressed Current* Berbasis Arduino UNO Dengan Tampilan LCD” ini, dibagi menjadi 2 bagian yaitu:

1. Pembuatan perangkat keras (*hardware*)

Pada tahap pertama ini meliputi semua proses pembuatan perangkat keras untuk merealisasikan hasil dari rancangan yang telah dibuat menjadi sistem yang siap dioperasikan.

2. Pembuatan perangkat lunak (*software*)

Tahap kedua ini yaitu mencakup semua hal yang berkaitan dengan perangkat lunak bagi sistem.

Pada masing-masing bagian mempunyai tujuan yang sama yaitu agar kedua bagian yang merupakan satu kesatuan sistem yang akan dibuat dapat saling melengkapi satu sama lain sehingga tercipta suatu sistem yang baik. Langkah awal dari pembuatan alat Tugas Akhir ini adalah perencanaan yang matang dan konsep yang jelas tentang aplikasi apa yang akan dibuat, agar kendala-kendala yang tidak diinginkan pada proses pembuatan dapat diperhitungkan terlebih dahulu.

4.1 Proses Pembuatan Perangkat Keras (hardware)

Pembuatan perangkat keras meliputi 2 bagian, yaitu pembuatan perangkat elektronika dan mekanik.

1. Pembuatan perangkat elektronika










Pembuatan perangkat elektronika ini merupakan pembuatan semua sistem yang berkaitan dengan perakitan elektronika yang meliputi perencanaan rangkaian, percobaan sementara, pembuatan rangkaian, serta pemasangan komponen.

2. Pembuatan bagian mekanik

Yang meliputi perencanaan bagian mekanik, pembuatan kerangka (tempat) untuk rangkaian, perakitan modul rangkaian pada kerangka (tempat) rangkaian, dan pembuatan label petunjuk penggunaan.

Pada pembuatan perangkat keras (*hardware*) ini dibutuhkan peralatan dan bahan-bahan untuk mendukung proses pengerjaannya. Tabel 4.1 dan 4.2 menunjukkan daftar alat dan bahan-bahan yang dibutuhkan dalam membuat alat Tugas Akhir ini.

Table 4.1 Daftar Alat Pembuatan Bagian Hardware

| NO. | NAMA | GAMBAR | SPESIFIKASI | JUMLAH |
|-----|---------------------------|---|---------------------|-------------------------|
| 1. | Pensil |  | Staedler 2B | 1 buah |
| 2. | Pengapus Pensil |  | Staedler 2B | 1 buah |
| 3. | Spidol Permanent |  | Snaowman G-12 | 1 buah (Warna Merah) |
| 4. | Penggaris |  | Mika Butterfly | 1 buah (Ukuran 30cm) |
| 5. | Multimeter |  | Digital Sanwa | 1 buah |
| 6. | Solder dan Blower |  | YiHUA 899D + 2 in 1 | 1 set |
| 7. | Pasta Solder |  | Lotfett | 12 gram |
| 8. | Atraktor (penyedot timah) |  | EDEN | 1 buah |
| 9. | Tang Pengupas Kabel |  | Nakai Model V | 1 buah |

Lanjutan Tabel 4.1





| NO. | NAMA | GAMBAR | SPESIFIKASI | JUMLAH |
|-----|------------------|---|---|--------|
| 10. | <i>Cutter</i> |  | Kenko L500 | 1 buah |
| 11. | Obeng Kombinasi |  | Dual Chrome Vanadium Baja | 1 Set |
| 12. | Kunci Ring Pas |  | Krisbow 6-22 mm (12pcs) | 1 set |
| 13. | Mesin Bor Tangan |  | KENMASTER Mesin Bor Bolak Balik + Mata Bor Besi Kayu Beton + Sekrup | 1 set |
| 14. | Gergaji Besi |  | Type HL-108 12inch | 1 buah |
| 15. | Mesin Gerinda |  | BOSCH GWS 060 Gerinda Tangan 4" | 1 buah |

Table 4.2 Daftar Bahan-Bahan Pembuatan Bagian Hardware

| NO. | NAMA | GAMBAR | SPESIFIKASI | UKURAN |
|-----|---------------------|---|--|---------------------------------------|
| 1. | Besi Tabung |  | Galvanis | Tebal 3 mm |
| 2. | Batu Gerinda Potong |  | WD | 1 Pak Isi 20 Pcs |
| 3. | Kayu Triplek |  | Multiplek | Ukuran 4 mm |
| 4. | Sekrup |  | Sekrup Kayu | Secukupnya |
| 5. | Spiral Kabel |  | KSS KS-10 | Secukupnya |
| 6. | Kabel |  | Kabel Pelangi | Secukupnya |
| 7. | PIN Header |  | Konektor <i>Female</i> dan <i>male</i> | Secukupnya |
| 8. | Timah Solder |  | Tenol Paragon 100 M 0,8mm | Secukupnya |
| 9. | Amplas |  | Amplas Serbaguna | Secukupnya |
| 10. | Box |  | Box Akrilik untuk Rangkaian | Ukuran: 19x12x7 cm & 40x8x10 cm |

4.1.1. Pembuatan Perangkat Elektronika

Pembuatan perangkat elektronika terdiri atas beberapa langkah yaitu:

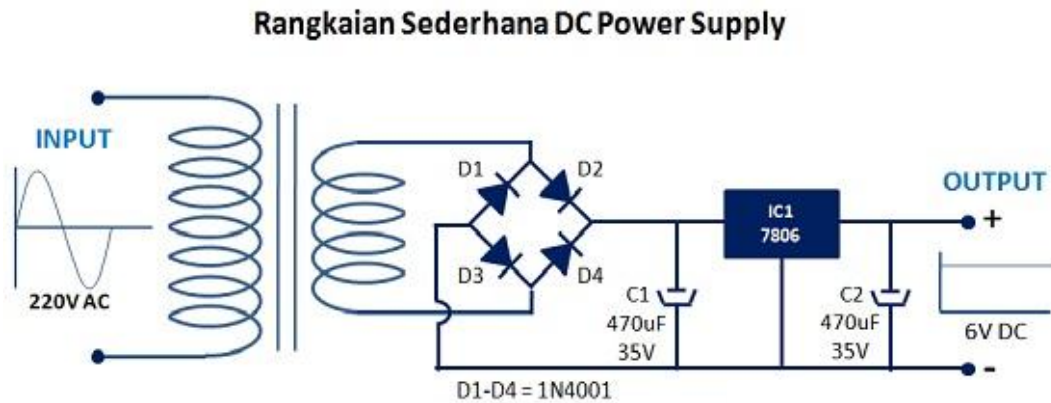
4.1.1.1. Perencanaan Rangkaian

Perencanaan rangkaian dilakukan untuk mendapatkan rangkaian sesuai dengan yang dibutuhkan, dengan mencari data-data tentang prinsip dasar dari komponen utama dan komponen bantu yang akan digunakan dalam rangkaian serta menentukan komponen-komponen yang akan digunakan. Setelah itu membuat gambar skema rangkaian baik untuk per-modul ataupun rangkaian sistem secara keseluruhan.

Pembuatan bagian elektronika terdiri atas beberapa langkah yaitu perencanaan rangkaian, percobaan sementara, pembuatan rangkaian pada papan *Printed Circuit Board* (PCB), serta pemasangan komponen. Dalam perencanaan ini terdapat dua rangkaian utama, yaitu:

4.1.1.1.1. Rangkaian Catu Daya

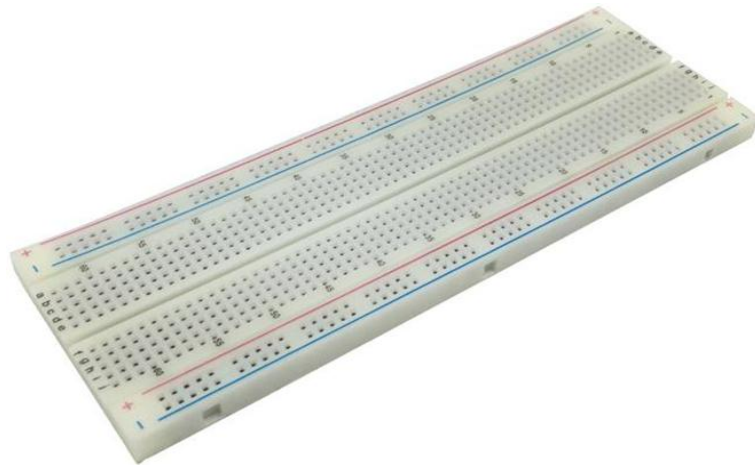
Rangkaian ini digunakan sebagai sumber untuk men-*supply* tegangan ke seluruh rangkaian sistem yang membutuhkan sumber tegangan sebesar 12 VDC dan 24 VDC. Gambar 4.1 di bawah ini menunjukkan rangkaian catu daya 12V dan 5V.



Gambar 4.1 Rangkaian Catu daya

4.1.1.2. Percobaan Sementara

Dalam pembuatan system ataupun rangkaian pada Tugas Akhir ini memerlukan rangkaian yang sesuai dan benar dengan spesifikasi yang diinginkan, sehingga rangkaian dan pengujian terlebih dahulu untuk mendapatkan rangkaian tersebut. Setelah rangkaian dirancang sesuai dengan skema, rangkaian tersebut perlu diuji coba untuk mengetahui apakah rangkaian dapat bekerja sesuai dengan apa yang diinginkan. Percobaan dilakukan dengan menggunakan papan rangkaian percobaan (*proto board/project board*) terlebih dahulu.

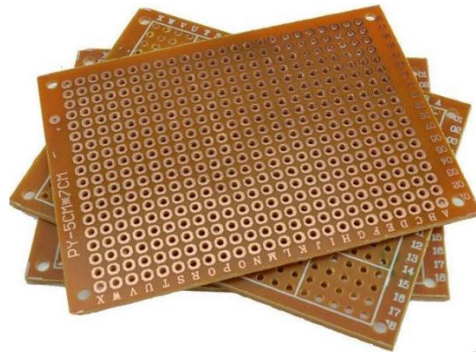


Gambar 4.2 Proto Board (Project Board)

Proses ini dilakukan agar jika terjadi kesalahan pada pemasangan komponen atau hasil keluaran tidak sesuai dengan keinginan, komponen dapat diganti dengan mudah. Setelah rangkaian dan hasilnya sesuai dengan yang dikehendaki, baru dibuat pola Papan Rangkaian.

4.1.1.3. Pembuatan Rangkaian

Langkah selanjutnya setelah percobaan rangkaian sistem adalah pembuatan rangkaian sesungguhnya setelah mengetahui susunan rangkaian yang benar. Dalam langkah ini digunakan sebuah papan sirkuit untuk tempat penyambungan agar antara komponen dapat terhubung sesuai gambar rangkaian yang telah dibuat sebelumnya. Papan sirkuit yang digunakan adalah *printed circuit board* (PCB).



Gambar 4.3 Printed circuit board PCB

Dengan menggunakan jenis PCB ini pekerjaan dalam pembuatan alat Tugas Akhir ini akan lebih mudah dan membantu untuk menghubungkan rangkaian yang telah di buat sebelumnya. Karena PCB sebagai tempat penyusunan komponen pembuatan alat *System Monitoring Impressed Current Cathodic Protection*.

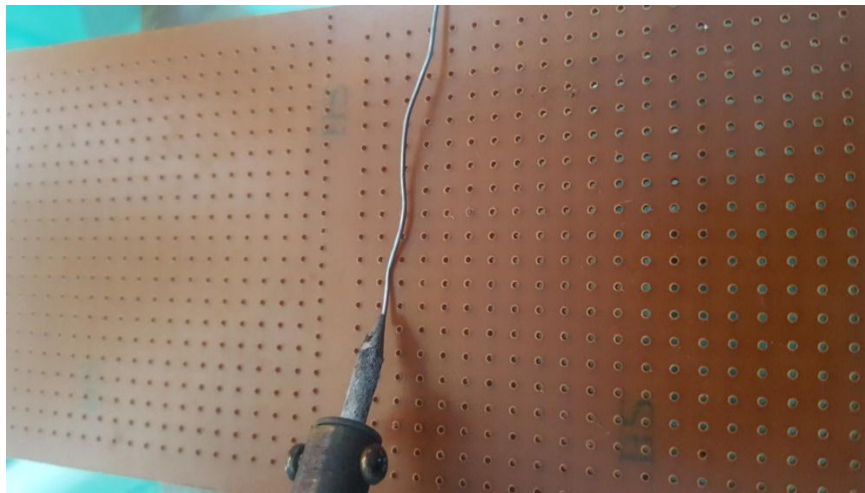
4.1.1.4. Pemasangan Komponen

Tahapan terakhir adalah memasang komponen yang telah dibuat oleh penyusun. Pemasangan di lakukan agar rangkaian yang telah dibuat, menjadi sebuah modul yang dapat digunakan di bagian proyek Tugas Akhir yang sedang dikerjakan.

Di antara pemasangan adalah untuk memasang catu daya (*power supply*) dan komparator yang akan digunakan. Berikut adalah langkah-langkah pemasangannya:

4.1.1.4.1. Pemasangan Komponen Catu Daya

- a. Memasang dan menyolder komponen-komponen aktif, yaitu diode, IC regulator, kapasitor dan terminal. Perlu diperhatikan bahwa pada saat pemasangan komponen ini posisi kaki-kakinya tidak boleh tertukar atau salah posisi pada PCB lubang. Gambar (4-5) di bawah ini menunjukkan proses pemasangan dan penyolderan komponen pada PCB lubang.



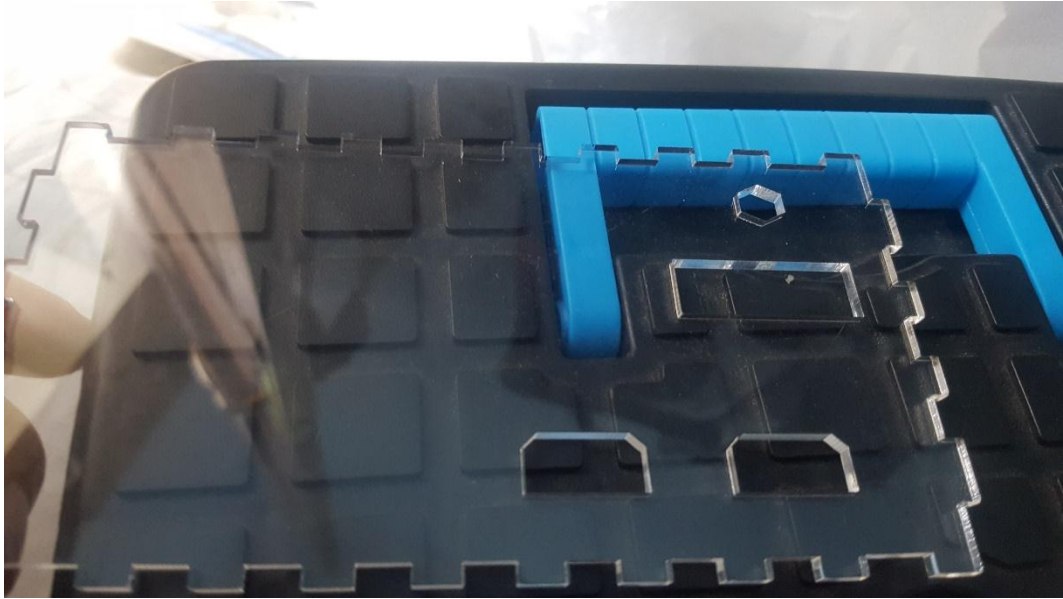
Gambar 4.4 Pemasangan dan Penyolderan Komponen Aktif

- b. Penyambungan komponen dioda dengan transformator 5 A, dimana 12 V pada trafo untuk rangkaian 12 V DC dan 5 V pada trafo untuk rangkaian 5VDC, serta 0 sebagai *ground* untuk rangkaian 12 V DC dan 5 V DC. Gambar (4-6) di bawah ini menunjukkan proses penyambungan / penyolderan rangkaian dengan transformator.



Gambar 4.5 Penyambungan dengan Transformator

- c. Pemasangan semua komponen dan transformator pada *box*, sebagai modul *power supply*. Diukur terlebih dahulu untuk letak papan PCB dan trafo agar tertata rapih.
- d. Membuat lubang dan memasang jack lubang pada power supply box. Pembuatan lubang untuk jack dilakukan dengan menggunakan bor ukuran 5mm. Pembuatan jack ini digunakan agar mudah dalam penggunaan. Gambar (4-9) di bawah ini menunjukkan proses pengeboran box catu daya sebagai tempat jack serta pemasangan jack pada box catu daya.



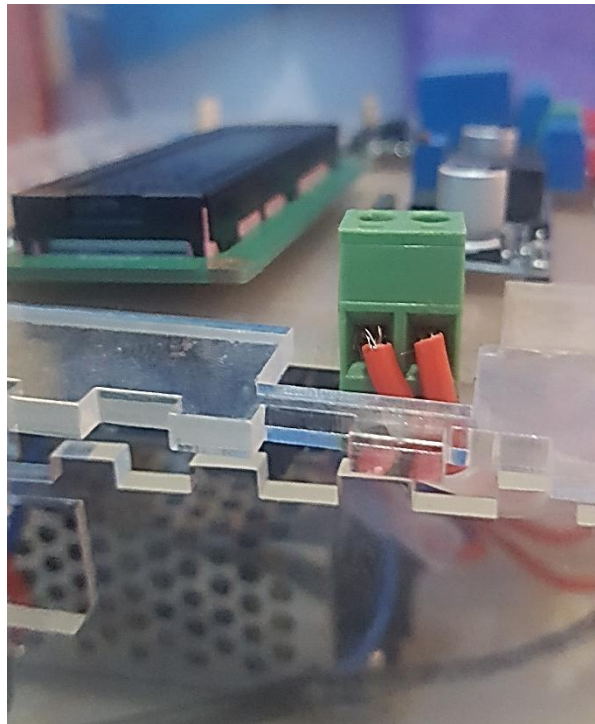
Gambar 4.6 Membuat Lubang dan Pemasangan Jack

- e. Pemasangan kabel *input* untuk power supply. Gambar 4.7 di bawah ini menunjukkan proses pemasangan kabel jack sebagai input tegangan 220VAc untuk rangkaian catu daya.



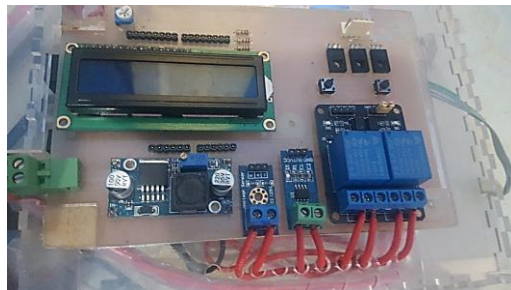
Gambar 4.7 Pemasangan Kabel Input

- f. Pemasangan kabel *output* untuk Arduino UNO. Gambar 4.8 di bawah ini menunjukkan proses pemasangan kabel jack sebagai input tegangan 12 V DC untuk rangkaian catu daya.



Gambar 4.8 Pemasangan Kabel Input Arduino UNO

- g. Pemasangan kabel *output* untuk sensor arus, sensor tegangan, dan relay. Gambar 4.9 di bawah ini menunjukkan proses pemasangan kabel jack sebagai input.



Gambar 4.9 Pemasangan Kabel output pada sensor arus, sensor tegangan dan relay

Pada pembuatan perangkat *Power Supply* ini dibutuhkan peralatan dan bahan-bahan untuk mendukung proses pengerjaannya. Tabel 4.3 menunjukkan daftar alat dan bahan-bahan yang dibutuhkan dalam membuat alat Tugas Akhir ini.

Tabel 4.3 Komponen *Power Supply*

| NO. | KOMPONEN | SPESIFIKASI | JUMLAH/UKURAN |
|-----|------------------|--------------------------------|---------------|
| 1. | Transformator | Merk “King” 5 A | 1 buah |
| 2. | Dioda | 3 A | 4 buah |
| 3. | Kapasitor | 1000 μ F / 35 V | 2 buah |
| 4. | IC Regulator | 7812 dan 7805 | 2 buah |
| 5. | Jack | banana dan lubang | 3 paket |
| 8. | Kabel tunggal | 0.75 mm | 2 meter |
| 9. | Kabel serabut | 1 mm | 1 meter |
| 10. | Kabel jack | H03VVH2-F 2x0.5mm ² | 1 buah |
| 11. | Power supply box | silver | 1 buah |
| 12. | Mur/baut halus | 2mm | 4 buah |
| 13. | <i>Spacer</i> | 1 cm | 4 buah |
| 14. | Baut kasar | 1 cm | 4 buah |

4.1.1.4.2. Pembuatan PCB DESIGN

Dalam pembuatan rangkaian komparator ada beberapa langkah, yaitu:

- a. Menyiapkan PCB polos untuk menentukan ukuran desain komponen – komponen yang akan di pasang.
- b. Buat *design* di *eagle*
- c. Cetak *design* komponen pada PCB yang telah disiapkan

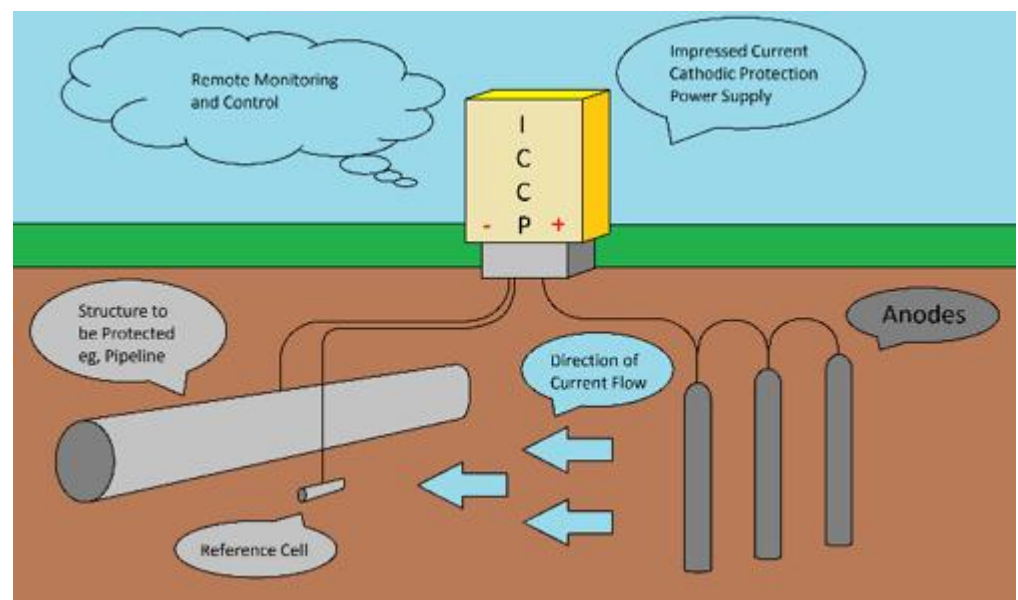
- d. Lubangi bagian yang perlu di lubangi untuk memudahkan komponen bisa menyatu dengan PCB.
- e. Sesuaikan komponen pada letak masing-masing *Single Line Diagram* yang sudah di buat.
- f. Setelah menyesuaikan pin di tempat yang sudah ditentukan, solder pin-pin tersebut untuk menyatukan komponen dengan komponen yang lainnya, agar saling terhubung.

4.1.2. Pembuatan Bagian Mekanik

Pembuatan bagian mekanik ini terdiri atas beberapa langkah yaitu:

4.1.2.1. Perencanaan Bagian Mekanik

Dalam pembuatan bagian ini penyusun terlebih dahulu harus membuat sketsa (*desain*) perencanaan kerangka yang sesuai dengan sistem mekanik pada proyek Tugas Akhir ini.

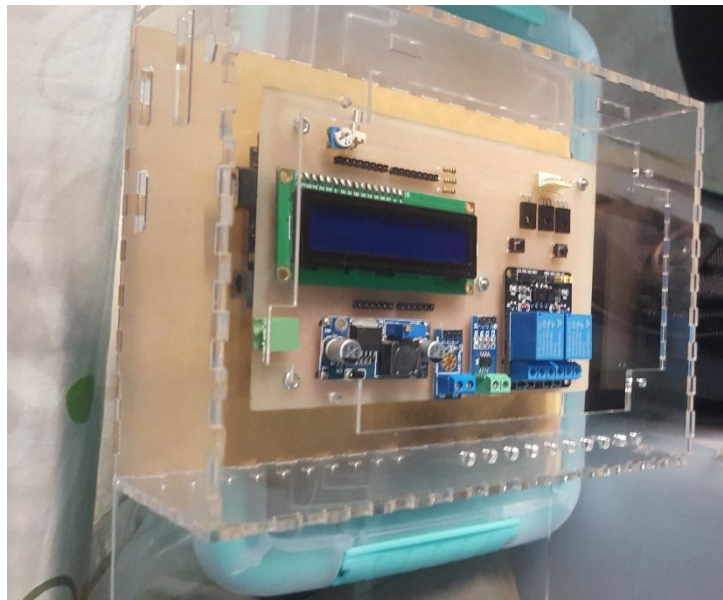


Gambar 4.10 Sketsa Mekanik

4.1.2.2. Pembuatan Kerangka (Tempat) Untuk Rangkaian

Pada tahapan ini penyusun harus membuat wadah untuk penempatan rangkaian komponen yang dipergunakan dalam bagian alat Tugas Akhir ini di antaranya adalah:

1. Membuat wadah pipa dari bahan akrilik.
2. Membuat dudukan berbentuk persegi panjang dari bahan kayu, sebagai alas penyatu komponen.



Gambar 4.11 Kerangka keseluruhan Mekanik

4.1.2.3. Perakitan Modul Rangkaian Pada Kerangka (Tempat) Rangkaian

Setelah rangkaian melalui tahap perakitan, pengukuran dan juga percobaan, kemudian langkah selanjutnya adalah memasangnya pada kerangka rangkaian yang telah dibuat, yaitu :

4.1.2.4. Pembuatan Label Petunjuk Penggunaan

Untuk memudahkan pengoperasian sistem diperlukan pemberian label fungsi. Langkah-langkah pemberian label fungsi adalah sebagai berikut:

1. Tempelkan label fungsi dan petunjuk pada tiap-tiap panel, terminal, serta modul-modul rangkaian dengan menggunakan huruf gosok/*rugos*.
2. Lapsi bagian permukaan kotak yang telah diberi label tadi dengan cat semprot transparan, atau jika tidak memungkinkan bisa gunakan isolasi transparan agar label atau tulisan tidak mudah terlepas.
3. Tempelkan petunjuk teknis penggunaan alat untuk menginformasikan cara penggunaan yang baik dan benar pada perancangan sistem keamanan pintu garasi, sehingga tidak terjadi kerusakan alat dikarenakan penggunaan yang tidak sesuai dengan langkah dan cara kerja alat.

4.2. Pembuatan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan penyusun dalam alat Tugas Akhir Perancangan sistem keamanan dengan menjadikan tegangan dan arus sebagai parameter untuk menghidupkan alarm dan mempermudah kerja operator dalam pemeliharaan alat. System ini dibuat menggunakan bahasa pemrograman C melalui aplikasi Arduino. Pada aplikasi Arduino bahasa C merupakan perangkat lunak yang menjadi bagian dari sistem yang mempunyai prinsip kerja mengatur kerja dari mikrokontroler Arduino Uno

dan keseluruhan perangkat keras (*hardware*) yang dihubungkan dengan mikrokontroler Arduino Uno itu sendiri.

Pada pembuatan Tugas Akhir ini dalam penggunaan perangkat lunak (*Software*) pada sistem sangat penting, karena perangkat lunak digunakan untuk mengatur prinsip kerja dari keseluruhan kerja sistem dari alat tersebut, baik perangkat keras maupun perangkat lunak itu sendiri.

Langkah-langkah pembuatan program sebagai berikut:

1. Membuat diagram alir (*flow chart*) dari program yang akan dibuat terhadap alat.
2. Membuat program dari sistem dengan menggunakan pemrograman bahasa C melalui aplikasi Arduino dengan referensi dari flowchart yang telah dibuat.
3. Mengkompilasi program yang telah dibuat hingga benar dan dapat digunakan, sehingga tidak terjadi kesalahan maupun error saat aplikasi di jalankan.
4. Pengisian program.

4.2.1. Pembuatan Diagram Alir (*Flowchart*)

Dalam menyusun diagram alir (*flowchart*) dari prinsip kerja alat, diusahakan dapat membagi proses dari bentuk kompleks menjadi sub program yang lebih kecil, sehingga lebih mudah dalam pembacaan dan jika ada kesalahan maka dalam pencarian kesalahan tersebut akan lebih mudah. Selain itu, dengan mempermudah dalam pemrograman maka dapat memudahkan orang lain dalam membaca alir program yang dibuat.

4.2.2. Pembuatan Program

Penulisan program dilaksanakan setelah diagram alur flowchart selesai dirancang. Pada pembuatan program mikrokontroler, diperlukan suatu sistem program untuk menempatkan dan mengirim program dari Personal *Computer* atau PC ke mikrokontroler.

4.2.2.1. Program Arduino UNO

Dalam pembuatan program Mikrokontroller dari Arduino UNO ini, perangkat lunak yang digunakan adalah aplikasi Arduino. Langkah kerja untuk membuat program yaitu:

1. Menyiapkan alat dan bahan yang diperlukan.
2. Kemudian sambungkan:
 - a. Kabel USB dari ke PC/komputer ke Arduino Uno.
 - b. Kaki 1 dan 16 terhubung dengan Ground (GND)
 - c. Kaki 2 dan 15 terhubung dengan VCC (+5V)
 - d. Kaki 3 dari LCD 16×2 adalah pin yang digunakan untuk mengatur kontras kecerahan LCD. Jadi bisa memasang sebuah trimpot 103 untuk mengatur kecerahannya. Pemasanganya seperti terlihat pada rangkaian tersebut, karena LCD akan berubah kecerahannya jika tegangan pada pin 3 ini di turunkan atau dinaikan.
 - e. Pin 4 (RS) dihubungkan dengan pin mikrokontroler
 - f. Pin 5 (RW) dihubungkan dengan GND
 - g. Pin 6 (E) dihubungkan dengan pin mikrokontroler
 - h. Sedangkan pin 11 hingga 14 dihubungkan dengan pin mikrokontroler sebagai jalur datanya.

- i. Jumper PIN kaki 5.0V Arduino Uno ke kaki Input Vcc rangkaian Signal Conditioning.
 - j. Jumper PIN kaki GND Arduino Uno ke kaki Input GND rangkaian Signal Conditioning.
 - k. Jumper PIN kaki 3 Arduino Uno ke PIN Trigger Sensor Ultrasonik HC-SR04.
 - l. Jumper PIN kaki 4 Arduino Uno ke PIN Echo Sensor Ultrasonik HC-SR04.
3. Buka software Arduino untuk memulai program:
- a. Klik kanan pada icon Arduino lalu pilih open atau bisa juga double klik



pada icon Arduino.

Gambar 4.12 Tampilan Software Aduino

Gambar 4.14 Tampilan untuk mengisi program Software Aduino

4. Kemudian, ketik program yang sudah siapkan di software tersebut.

Penampilan di bawah ini menunjukkan masing-masing tampilan pembacaan dari program dengan menggunakan Mikrokonroller Arduino Uno secara keseluruhan.

```

//===== Import library=====
#include <LiquidCrystal.h> //Library untuk LCD Liquid
Crystal
#include <TimerOne.h> //Library digunakan untuk timer
interrupt

//===== Inisialisasi =====
LiquidCrystal lcd(13, 12, 11, 10, 9, 8); //Penggunaan pin
pada LCD Liquid Crystal
float A1read = 0.0, sampling = 0.0, average = 0.0, value_mA
= 0.0; //Pin digunakan untuk membaca sensor ACS (Analog pin
1)
int A0read = 0, vin = 0; //Pin digunakan untuk membaca
sensor tegangan (Analog pin 0)
int buttonPin1 = 3, buttonPin2 = 4, relayPin1 = 5, relayPin2
= 6; //buttonPin digunakan sebagai masukan, dan relayPin
digunakan sebagai keluaran
bool bMaintenance = false; //Digunakan untuk tanda (flag)
apakah dalam kondisi maintenance atau tidak
int relayState1 = HIGH, relayState2 = HIGH; //Konsisi relay
int buttonState1, buttonState2; //Kondisi pushbutton
int lastButtonState1 = LOW, lastButtonState2 = LOW;
//Konsisi terakhir pada push button

```

```

unsigned long lastDebounceTime1 = 0, lastDebounceTime2 = 0;
//Menyimpan nilai penggunaan waktu debouncing

///===== Inisialisasi Fungsi =====
void buttonFunct(void); //Fungsi pembacaan tombol dengan
debounce (50ms)
void currentSensor(void); //Fungsi pembacaan sensor arus ACS
void voltageSensor(void); //Fungsi pembacaan sensor tegangan
void lcdSetupDisp(void); //Fungsi setting display LCD awal
(pesan)
void errorFunct(int port, int port1, int hValue, int
lValue); //Fungsi penampil pesan/konsisi error
void maintenanceFunct(void); //Fungsi saat kondisi
maintenance

///===== Fungsi main =====
void setup() {
    ///===== Inisialisasi penggunaan pin/mode pin =====
    pinMode(0, OUTPUT); digitalWrite(0, LOW); //LED Kuning
    pinMode(1, OUTPUT); digitalWrite(1, LOW); //LED Hijau
    pinMode(2, OUTPUT); digitalWrite(2, LOW); //LED Merah
    pinMode(3, INPUT_PULLUP); //Pembacaan pada pin
pushbutton menggunakan PULLUP internal micro pin 3
    pinMode(4, INPUT_PULLUP); //Pembacaan pada pin
pushbutton menggunakan PULLUP internal micro pin 4
    pinMode(relayPin1, OUTPUT); digitalWrite(relayPin1,
HIGH); //Pin relay diset aktif tinggi (HIGH) dan untuk
keluaran pin 5
    pinMode(relayPin2, OUTPUT); digitalWrite(relayPin2,
HIGH); //Pin relay diset aktif tinggi (HIGH) dan untuk
keluaran pin 6

```

```

    pinMode(7, OUTPUT); digitalWrite(7, LOW); //Pin
digunakan untuk aktivasi buzzer dengan aktif rendah (LOW)
    pinMode(A0, INPUT); //Mode INPUT untuk pembacaan
sensor tegangan
    pinMode(A1, INPUT); //Mode INPUT untuk pembacaan
sensor arus (ACS)
    lcd.begin(16, 2); //Setting fungsi pada LCD dengan
16x2 (baris x kolom)
    lcdSetupDisp(); //Pemanggilan fungsi lceSetupDisp()
    Timer1.initialize(1000); //Inisialisasi fungsi
timerOne, dipanggil setiap 1ms sekali
    Timer1.attachInterrupt(buttonFunc); //Penggunaan
interrupt pada fungsi pembacaan pushbutton (buttonFunc)
}

//===== Fungsi loop =====
void loop() {
    buttonFunc(); //Memanggil fungsi pembacaan pushbutton
(buttonFunc())
    voltageSensor(); //Memanggil fungsi pembacaan sensor
tegangan (voltageSensor())
    currentSensor(); //Memanggil fungsi pembacaan sensor
arus (currentSensor())
    if (value_mA < 0.1) { //Apabila nilai arus (value_mA)
kurang dari 0.1mA, maka fungsi error akan dijalankan
        errorFunc(2, 7, 500, 500); //Memanggil fungsi
error, menyalakan LED merah pin 2, Buzzer pin 7, dan nilai
delay 500ms
    }
    else { //Apabila nilai if() tidak terpenuhi maka
mengekseskui program selanjutnya

```



```

        digitalWrite(1, HIGH); //Menyalakan LED hijau
        sebagai tanda perangkat berjalan normal (value_mA > 0,1mA)
    }
    if (bMaintenance == true) { //Apabila dalam konsisi
        maintenance, maka syarat boolean akan etrpenuhi dan
        mengeksekui program maintenanceFunct()
        maintenanceFunct(); //Memanggil fungsi
        maintenanceFunct dan menyalakan LED kuning
    }
    delay(2000); //Nilai delay dalam sekali looping
}
///===== Fungsi SetupLCD Liquid Crystal =====
void lcdSetupDisp(void) {
    lcd.setCursor(1, 0); //Set cursor pada column 1 dan
    row 0
    lcd.print("---WELCOME---"); //menampilkan pesan
    string
    delay(2000); //Nilai pewaktu delay dalam menampilkan
    pesan
    lcd.clear(); //Menghapus pesan di LCD
    lcd.setCursor(0, 0); //Set cursor pada column 0 dan
    row 0
    lcd.print("====CATHODE===="); //menampilkan pesan
    string
    delay(500); //Nilai pewaktu delay dalam menampilkan
    pesan
    lcd.setCursor(0, 1); //Set cursor pada column 1 dan
    row 0
    lcd.print("---MONITORING---"); //menampilkan pesan
    string

```

```

        delay(2000); //Nilai pewaktu delay dalam menampilkan
pesan
        lcd.clear(); //Menghapus pesan di LCD
    }
    ///===== Fungsi pembacaan sensor arus (ACS) =====
    void currentSensor(void) {
        for (int i = 0; i < 255; i++) { //Nilai looping dan
increment dengan flag/batas 0-255
            A1read = analogRead(A1); //Membaca nilai sensor
arus pada pin A0 (analog)
            sampling = sampling + A1read; //Menambah hasil
pembacaan sejumlah i dan menghasilkan nilai akhir berupa
akumulasi
        }
        average = (sampling / 255.0) - 509.0; //Mencari nilai
rata-rata dari hasil pembacaan sampling dibagi total i (255)
dikurangi pembacaan nilai analog 509.0
        A1read = ((average * (5.0 / 1023.0)) / (65 / 25)) *
10.0; //Hasil pembacaan kemudian dikali perbandingan nilai
tegangan refrensi dibagi total 1 byte,

                //Dan dibagi dengan nilai konstanta
sesuai spesifikasi sensor dari pabrik
        if (A1read < 0.030) { //Apabila nilai A1read kurang
daro 0.030 maka akan mengeksekusi program
            A1read = 0.0; //Menganggap pembacaan bernilai 0.0
jika pembacaan sebenarnya dibawah 0.030 (mengabaikan)
        }
        value_mA = A1read * 1000.0; //Menhkalikan nilai mA
dengan 1000 menjadikannya A

```

```

        lcd.setCursor(0, 1); //Mmpersiapkan posisi cursor
untuk menampilkan nilai arus
        lcd.print("i = "); //Menampilkan pesan
        lcd.print(value_mA, 1); //Menampilkan hasil pembacaan
pada LCD
        lcd.setCursor(8, 1); //Set posisi cursor pada LCD
        lcd.print(" mAmps"); //Menampilkan pesan
        //delay(2500);
        sampling = 0; //Me-reset nilai sampling kembali ke 0
    }
    ///===== Fungsi pembacaan sensor tegangan =====
    void voltageSensor(void) {
        float temp; //Inisaialisasi
        A0read = analogRead(A0); //Pembacaan niali data analog
pada pin A1
        temp = A0read / 4.092; //Hasil pembacaan kemudaian
dibandingkan dengan tegangan refrensi (5.0 volts)
        vin = (temp / 10); //Hasil perbandingan dibagi 10
untuk pembulatan nilai
        lcd.setCursor(0, 0); //Set cursor pada LCD
        lcd.print("v = "); //Menampilkan pesan
        lcd.print(vin, 1); //Menampilkan pesan pembacaan niali
tegangan
        lcd.setCursor(8, 0); //Set cursor pada LCD
        lcd.print(" volts"); //Menampilkan pesan
        //delay(2500);
    }
    ///===== Fungsi error =====
    void errorFuncnt(int port, int port1, int high, int low) {
        digitalWrite(port, HIGH); //Set keluaran menjadi aktif
tinggi (HIGH)

```

```

    digitalWrite(port1, HIGH);
    delay(high); //nilai delay sesuai parameter high
    digitalWrite(port, LOW); //Set keluaran menjadi aktif
rendah (LOW)
    digitalWrite(port1, LOW);
    delay(low); //nilai delay sesuai parameter high
}
//===== Fungsi pembacaan tombol =====
void buttonFunct(void) {
    int reading = digitalRead(3); //Membaca masukan tombol
pada pin 3
    int reading2 = digitalRead(4); //Membaca masukan
tombol pada pin 3
    if (reading != lastButtonState1) { //Jika pembacaan
tidak sama dengan kondisi set akhir
        lastDebounceTime1 = millis(); //Maka nilai
lastDebounceTime1 akan mencatat penggunaan waktu (ms)
    }
    if (reading2 != lastButtonState2) { //Jika pembacaan
tidak sama dengan kondisi set akhir
        lastDebounceTime2 = millis(); //Maka nilai
lastDebounceTime2 akan mencatat penggunaan waktu (ms)
    }
    if ((millis() - lastDebounceTime1) > 50) { //Apabila
waktu saat ini dikurangi penggunaan waktu saat pemencetan
tombol lebih dari 50ms maka program dieksekusi
        if (reading != buttonState1) { //Apabila
pembacaan tidak sama dengan kondisi buttn saat ini maka
program dieksekusi
            buttonState1 = reading; //Apabila tidak
sesuai, maka disesuaikan

```

```

        if (buttonState1 == HIGH) { //Jika kondisi
button aktif tinggi, maka akan mengupdate kondisi terakhir
keluaran (relay)

                relayState1 = !relayState1;
//Membalikkan keadaan aktif rendah atau aktif tinggi
        }
    }
}
    if ((millis() - lastDebounceTime2) > 50) { //Apabila
waktu saat ini dikurangi penggunaan waktu saat pemencetan
tombol lebih dari 50ms maka program dieksekusi
        if (reading2 != buttonState2) { //Apabila
pembacaan tidak sama dengan kondisi buttn saat ini maka
program dieksekusi
            buttonState2 = reading2; //Apabila tidak
sesuai, maka disesuaikan
                if (buttonState2 == HIGH) { //Jika kondisi
button aktif tinggi, maka akan mengupdate kondisi terakhir
keluaran (relay)

                        relayState2 = !relayState2;
//Membalikkan keadaan aktif rendah atau aktif tinggi
                }
            }
        }
        digitalWrite(5, relayState1); //Pin 5 sebagai output
dan ssuai kondisi relay yang diinginkan
        lastButtonState1 = reading; //Update hasil pembacaan
ke lastButtonState sebagai perbandingan pembacaan diatas
        digitalWrite(6, relayState2); //Pin 6 sebagai output
dan ssuai kondisi relay yang diinginkan

```

```
        lastButtonState2 = reading2; //Update hasil pembacaan
ke lastButtonState sebagai perbandingan pembacaan diatas
        if (relayState1 == LOW & relayState2 == LOW) {
//Apabila salah satu tombol ditekan, maka akan mengeksekusi
program
                bMaintenance = false; //Boolean untuk flag
kondisi maintenance
        }
        else bMaintenance = true; //Kondisi kebalikan sesuai
if
}
//===== Fungsi maintenance =====
void maintenanceFunct(void) {
        digitalWrite(0, HIGH); //Output sinyal digital pada
pin 0
        delay(500); //Delay selama nilai 500ms
        digitalWrite(0, LOW); //Output sinyal digital pada pin
0
        delay(500); //Delay selama nilai 500ms
}
//END OF PROGRAM
```