

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Dalam penulisan tugas akhir ini metode yang digunakan dalam penelitian adalah :

1. Metode Perancangan

Merupakan proses untuk membuat rancangan sistem yang digunakan sebagai objek penelitian yang dilakukan sampai pada hasil penelitian yang diharapkan.

2. Realisasi Perancangan

Merupakan proses untuk menjadikan suatu rencana menjadi perwujudan yang nyata.

3. Implementasi

Cara menerapkan perancangan.

4. Analisa

Analisa Perbandingan implementasi alat dengan perancangan alat, sehingga sesuai dengan kebutuhan sistem.

3.2 Tempat Penelitian

Penelitian dan Pengujian alat dilakukan di rangkaian PT.Putra Mulia Telekomunikasi Medan.

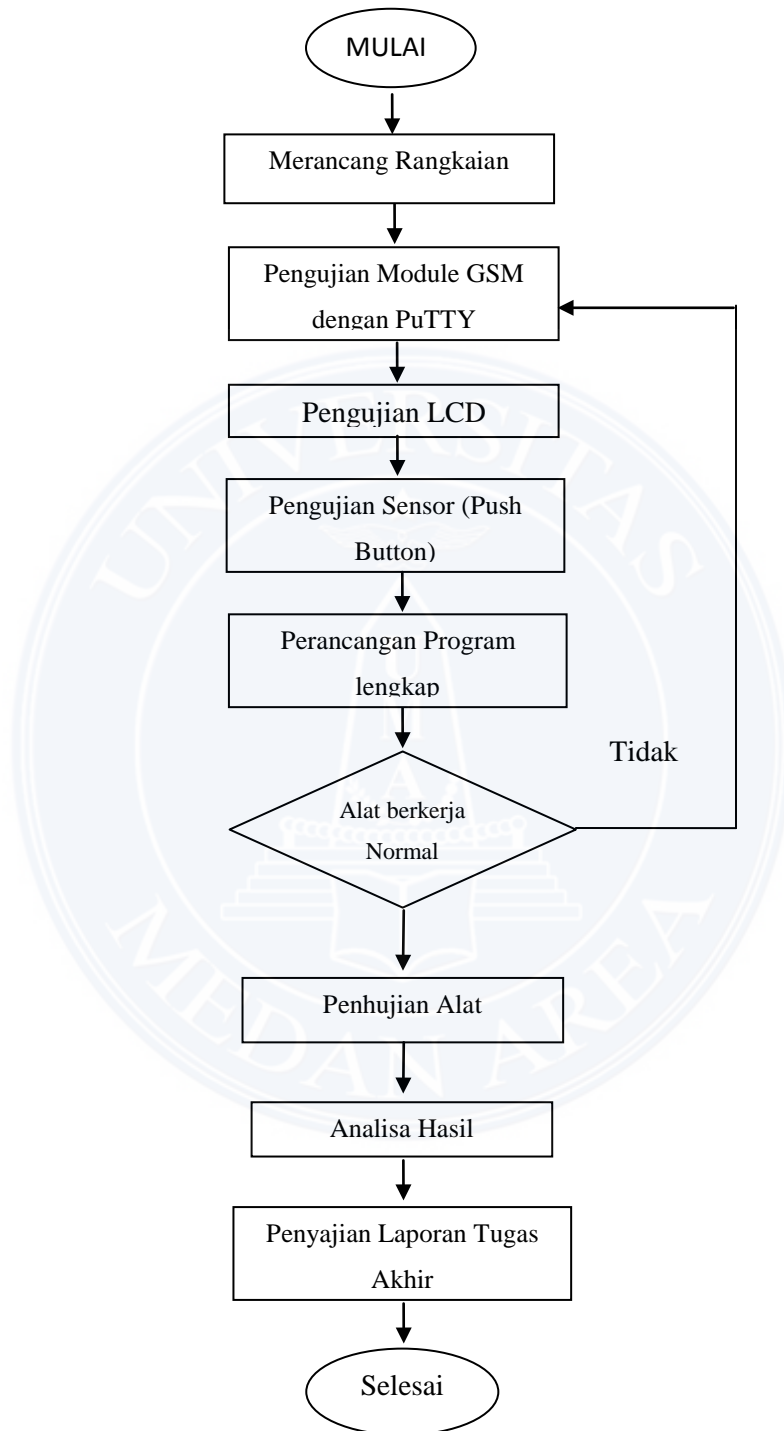
3.3 Peralatan Yang digunakan

Untuk mendapatkan hasil perancangan yang optimal , maka sangat dibutuhkan peralatan berikut:

- | | |
|------------------------------|------------|
| 1. Multitester Digital | 1 (unit) |
| 2. Downloader Mikrokontroler | 1 (unit) |
| 3. Laptop | 1 (unit) |
| 4. Kabel jumper | secukupnya |
| 5. Converter USB to Serial | 1(Unit) |

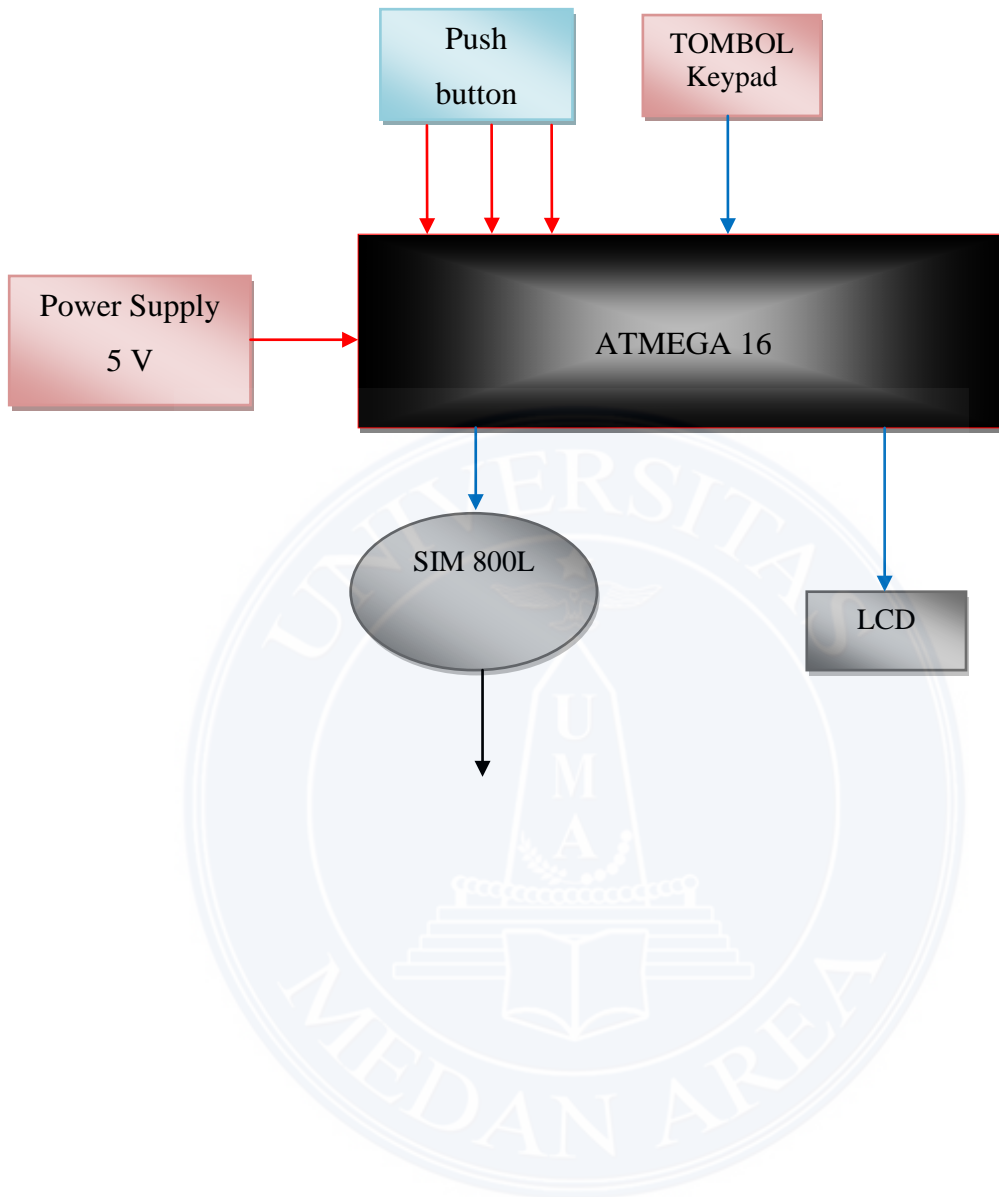
3.4 Flow Chart Penelitian

Adapun *Flow chart* pada penelitian ini adalah sebagai berikut



Gambar 3.1 *Flow Chart* Penelitian

3.5 Diagram Blok Perancangan



Gambar 3.2 Diagram Blok Rangkaian

Adapun fungsi masing – masing blok diagram diatas antara lain :

1. *Power Supply 5V*

Fungsi dari *power supply 5V* adalah mensuplai tegangan 5 Volt DC.dalam perancangan ini yang di suplai ialah rangkaian *mikrokontroller* , dan rangkaian LCD (*Liquid Crystal Display*)

2. Mikrokontroler Atmega 16

Fungsi dari *mikrokontroler* Atmega 16 adalah sebagai pengendali utama. komunikasi antar *Push button*, *Module SIM800L* dan komunikasi antar LCD.

3. Push Button

Fungsi dari *Push Button* adalah sebagai *trigger* alarm ke *mikrokontroler* Atmega 16 dan alarm tersebut akan di *transfer* melalui *Module SIM800L* ke *User*.

4. LCD (Liquid Crystal Display)

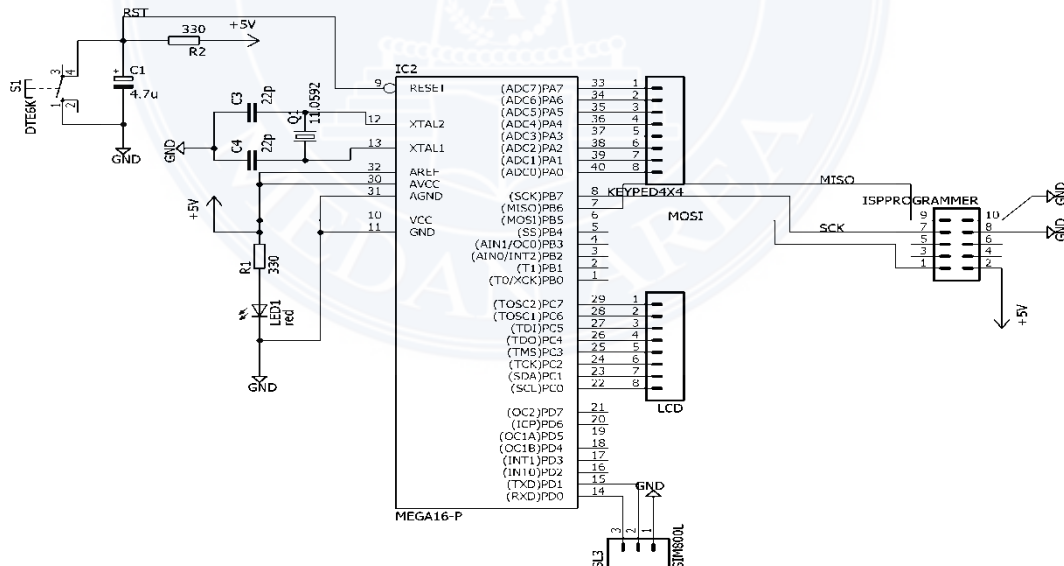
Fungsi dari LCD adalah menampilkan arus serta mengetahui apakah arus yang mengalir di AC sesuai atau tidak.

5. Tombol Pilihan (Keypad)

Fungsi dari tombol ini ialah *turn on* ataupun *turn off system*

3.6 Perancangan Perangkat Keras (Hardware)

3.6.1 Rangkaian Mikrokontroler Atmega 16



Gambar 3.3. Rangkaian Mikrokontroler ATmega 16

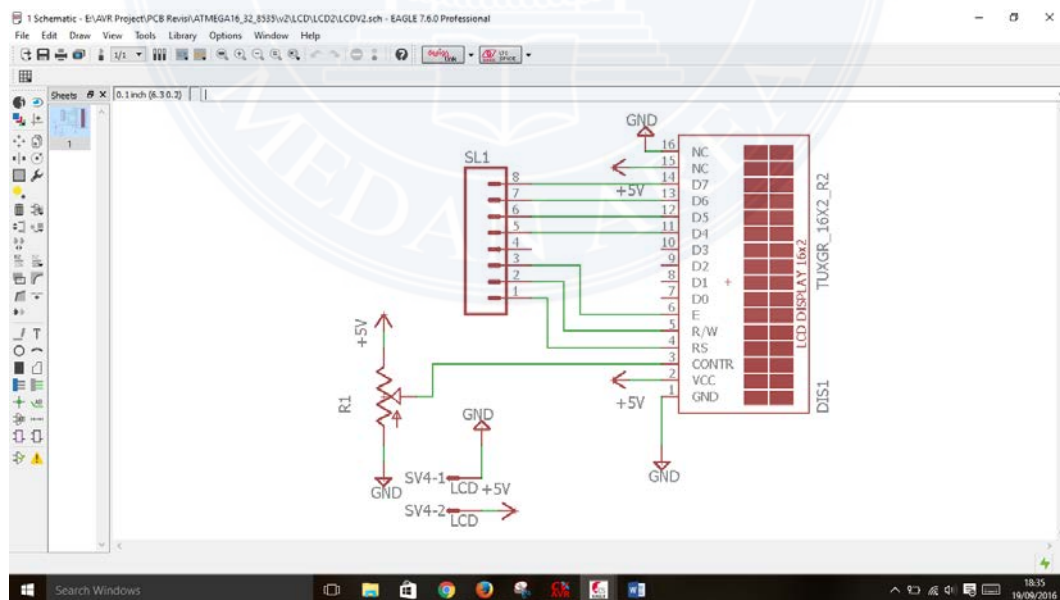
Rangkaian *mikrokontroler* ini merupakan tempat pengolahan data dan pengoperasian alat. Dan dalam rancangan ini, *mikrokontroler* berfungsi sebagai otak dari seluruh sistem rancangan. *Mikrokontroler* Atmega 16 ini memiliki 4 buah *port* dan berbagai pin yang digunakan untuk menampung *input* dan *output* data

dan terhubung langsung dengan rangkaian-rangkaian pendukung lainnya. *Port* yang akan digunakan dalam pembuatan antara lain:

1. PORTD.0 dan PORTD.1 digunakan sebagai tempat proses merubah sinyal analog menjadi digital yang berasal dari *Module SIM800L*
2. PORTC.0 sampai PORTC.7 digunakan sebagai komunikasi ke antar LCD.
3. Pin *reset* pada *mikrokontroler* Atmega 16 terletak pada Pin 9 Yang membentuk rangkaian *power on reset* di mana rangkaian ini akan mereset rangkaian *mikrokontroler*, sehingga *mikrokontroler* tersebut kembali menjalankan program yang ada di dalamnya dari awal.
4. PORTA.0 sampai PORTA.7 Digunakan untuk keypad *turn on/off* dan *set password* sistem
5. PORTB.5 dan PORTB.5 Digunakan untuk koneksi ke downloader.

3.6.2 Rangkaian LCD (*Liquid Crystal Display*)

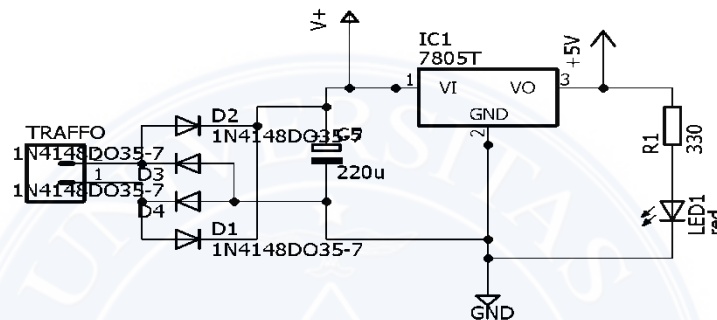
Pada tugas akhir ini, LCD digunakan untuk password dan alarm aktif pada media display. LCD yang digunakan adalah LCD 2x16 dengan tipe 1602ZFA dengan lebar *display* 2 baris dan 16 kolom. Hubungan antara *mikrokontroler* dan LCD diperlihatkan pada Gambar 3.5.



Gambar 3.4 Rangkaian LCD

Untuk Mengatur Kontras Pada LCD, Dipasang Potensiometer Dengan Besar tahanan antara 10k–100K Sebagai Pengatur kontras karakter. komunikasi Antara LCD Dengan *Mikrokontroler* Atmega 16 terletak pada pin yang telah ditentukan RS dan E dihubungkan ke PORTC.0 DAN PORTC.2 Dan Pin C11 Sampai C13 Pada LCD Dihubungkan Ke PORTC.4 Sampai PORTC.7 pada mikrokontroler.

3.6.3 Rangkaian Catudaya 5 VDC



Gambar 3.5. Rangkaian Catudaya 5 Volt

Rangkaian Catu Daya adalah rangkaian pengatur tegangan agar tegangan yang keluar dari rangkaian ini tetap pada satu nilai meskipun masukkannya lebih besar dari nilai yang diinginkan. Pada rancangan ini digunakan IC LM 7805 sebagai regulator tegangan, dikarenakan LM7805 bisa menerima tegangan masukan antara 7V-18V sesuai data sheetnya. tetapi tegangan keluarannya bernilai 5V yang Sesuai dengan tegangan yang dibutuhkan oleh *Mikrokontroler* , sensor arus serta LCD sebagai catu dayanya.

3.7 Perancangan Perangkat Lunak (Software)

3.7.1 Instalasi CodeVision AVR

Langkah – langkah Menginstal Software Code Vision AVR

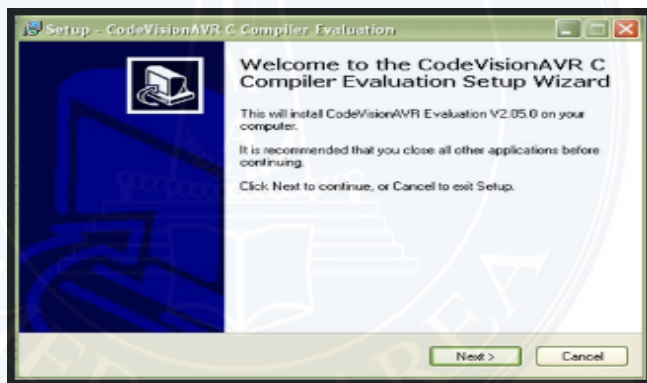
1. Buka folder software CodeVision AVR, kemudian cari File Setup.exe seperti gambar dibawah. kemudian buka (doble klick / klik kanan open)



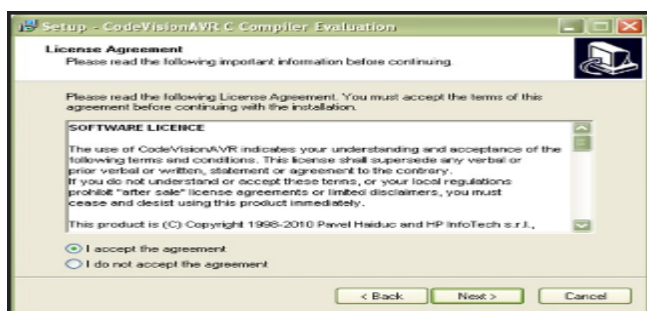
- Setelah dibuka maka akan tampil jendela seperti gambar dibawah, Jendela ini untuk pemilihan bahasa yang akan kita gunakan, jika kita mau menggunakan bahasa inggris maka pilih English kemudian klik OK.



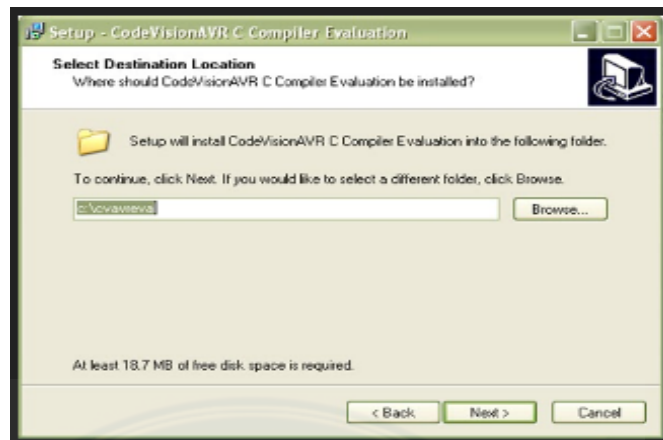
- Kemudian Klik Next



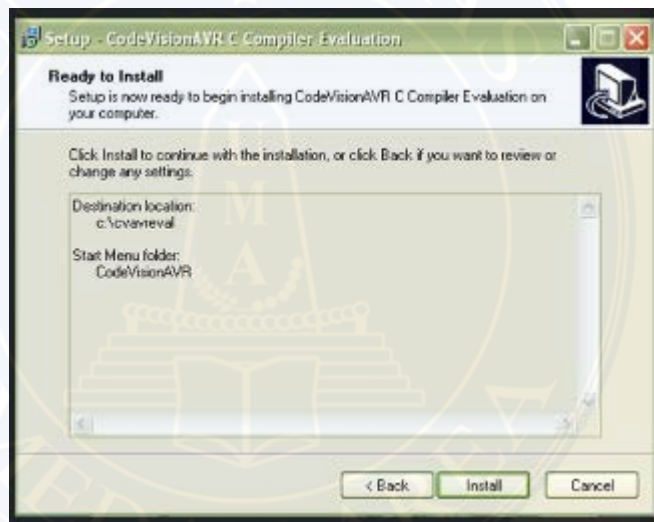
- Kemudian akan muncul jendela seperti dibawah kemudian pilih I accept the agreement dan klik Next



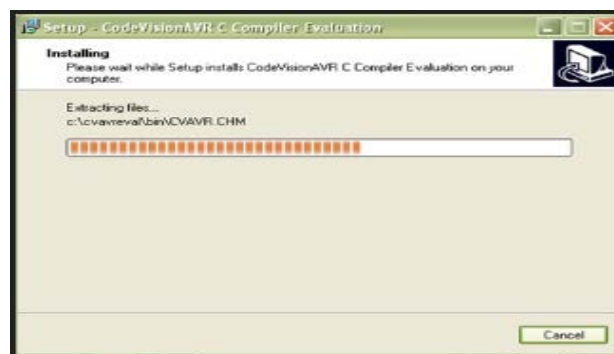
5. Klik Next



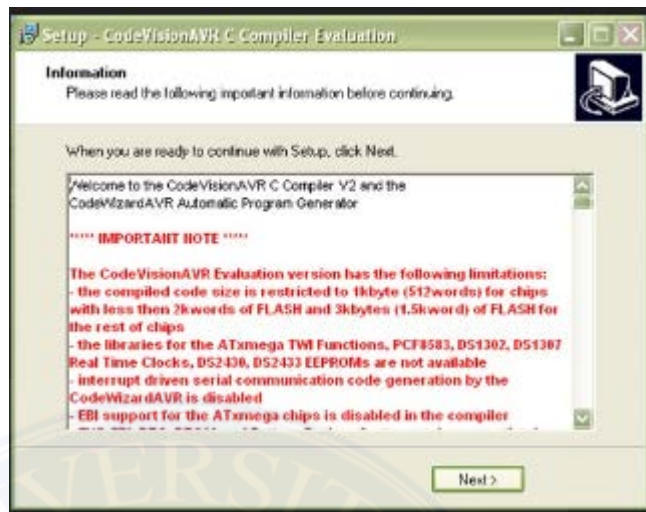
6. Klik Install



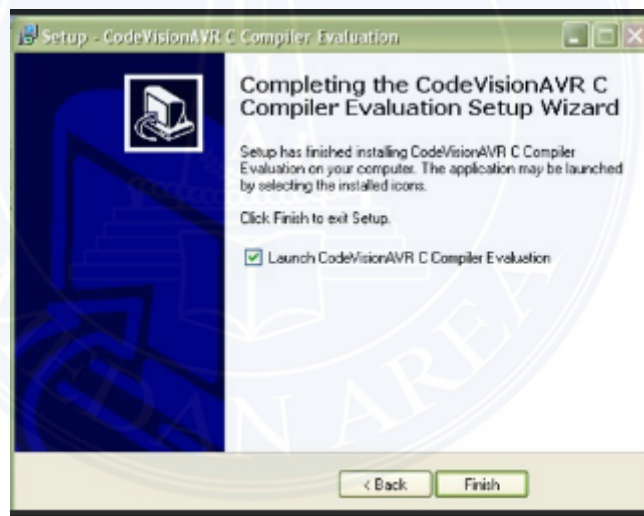
7. Maka akan tampil seperti gambar dibawah ini. Tunggu sampai proses selesai.



8. Kemudian Klik Next



9. Proses Instal Selesai.



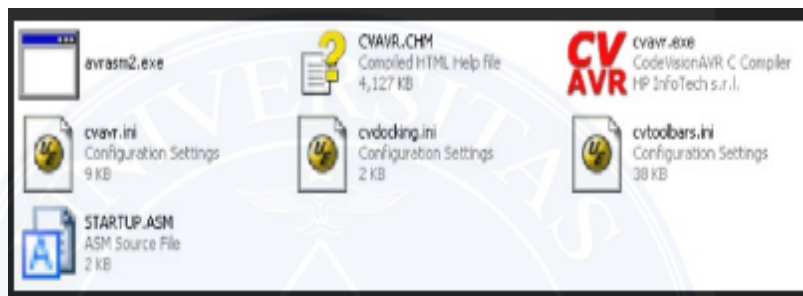
10. Proses selanjutnya kita akan mengubah versi demo dengan Full Version. Dengan menggunakan file chacknya (fatch). Caranya buka kembali folder mentahan software Codevision. Kemudian cari folder Crack dan buka. Ketika dibuka maka akan terdapat file cvavr.exe kemudian copy (klik kanan mouse copy / Ctrl+c).



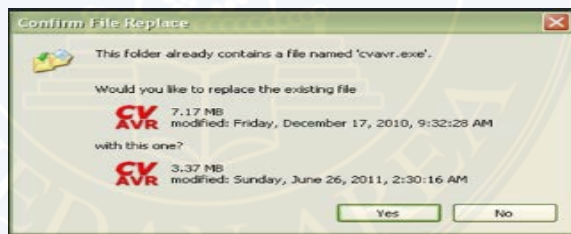
11. Setelah file dicopy langkah selanjutnya meletakkan file yang dicopy tadi ke lokasi folder bin dengan cara di *paste* (klik kanan *paste/Ctrl + V*).

Jika instalasi anda di letakan di drive c maka letak filenya berada di:

C:\cvavreval\bin



12. Setelah dipaste maka akan muncul jendela seperti berikut kemudian klik Yes.



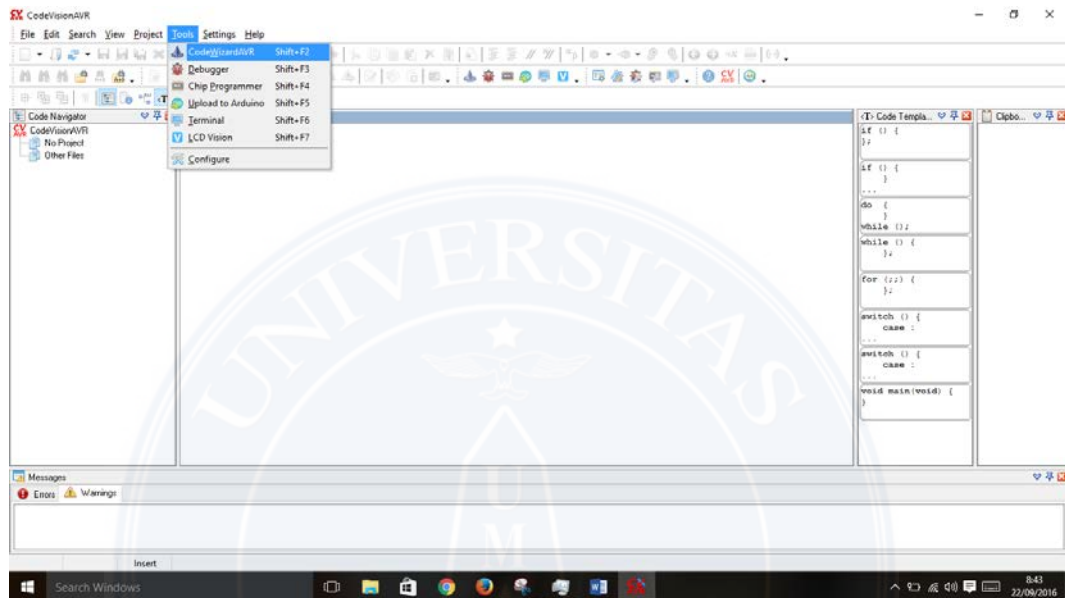
13. Proses instal selesai.. Jika anda membuka softwarenya maka akan tampil jendela awalnya seperti berikut ini.



Gambar 3.6 Proses Instalasi *CodeVision AVR*

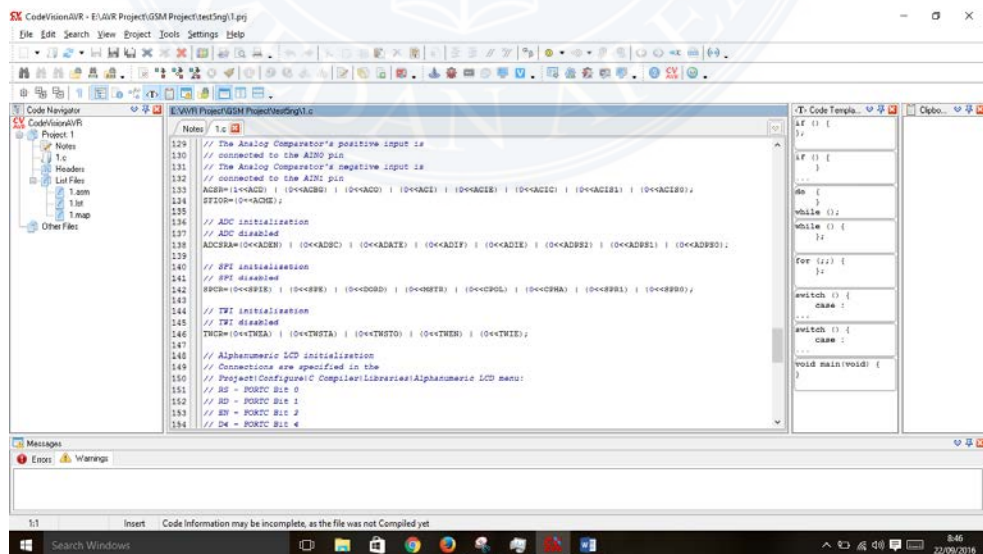
3.7.2 Membuat listing program dengan CodeVision AVR

Pada perancangan perangkat lunak yaitu menggunakan *software* CodeVision AVR yang digunakan untuk menuliskan *listing* program dan mengkompilasi *file* program menjadi *file* hexa. *File* hexa yang dihasilkan setelah proses kompilasi tersebut akan dimasukkan ke dalam *mikrokontroler*, sehingga *mikrokontroler* akan bekerja sesuai dengan perintah yang ada.



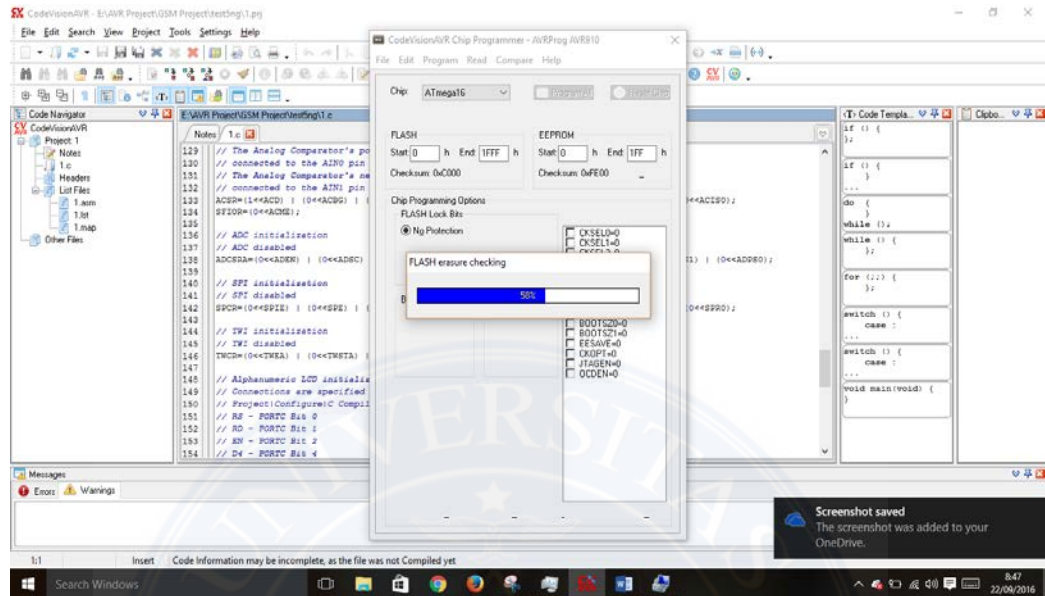
Gambar 3.7 Halaman utama CodeVision AVR

Setelah *form* utama program CodeVision AVR ditampilkan, maka selanjutnya adalah menuliskan *listing* program dapat dilihat pada gambar berikut ini.



Gambar 3.8 Contoh penulisan listing program

Langkah selanjutnya adalah mengkompile program, dengan cara memilih *icon Compile Program* atau tekan F9 pada *keyboard* agar listing program yang dibuat dikompile menjadi *file* dengan *extention hex*.



Gambar 3.9 Proses Meng*compile* Program

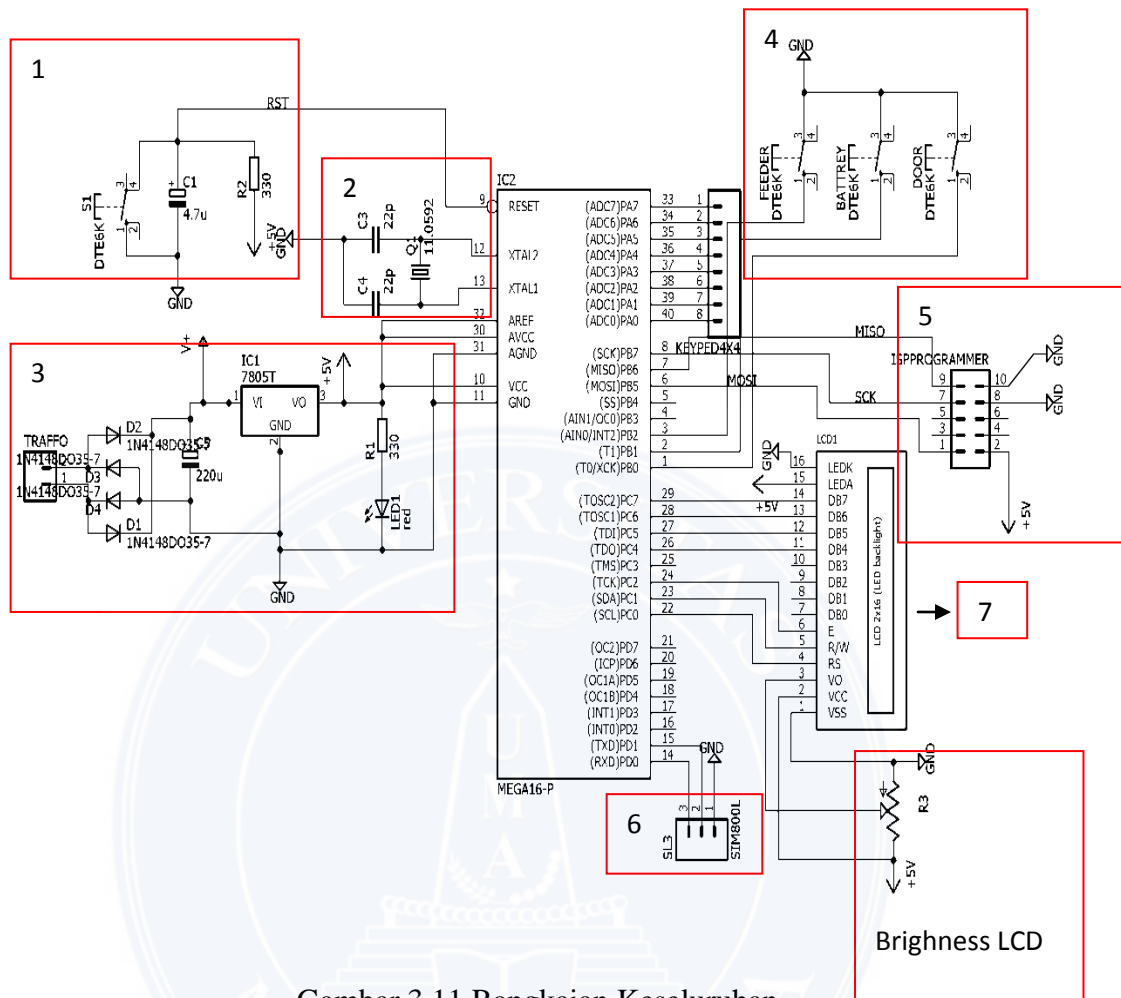
3.7.3 Mengisi Program Mikrokontroler Atmega 16

Mikrokontroler bisa bekerja jika di dalam sudah dimasukan listing program yang sudah dibuat dengan menggunakan software CodeVision AVR. Untuk melakukan proses pengisian program kedalam *mikrokontroler* Atmega 16 dibutuhkan perangkat Downloader USB ISP AVR (USB AVR910 Downloader)



Gambar 3.10 USB ISP AVR (USB AVR910 Downloader)

3.8 Rangkaian Keseluruhan



Gambar 3.11 Rangkaian Keseluruhan

1. Rangkaian Reset *Mikrokontroller*
2. Rangkaian *Crystal* untuk membangkitkan Clock (Sinyal Pulsa)
3. Catu Daya
4. Tombol tekan / *Switch Alarm*
5. *Interface USB Downloader*
6. Komunikasi Serial / *GSM 800L*
7. *LCD (Liquid Crystal Display)*

BAB IV

HASIL DAN PEMBAHASAN

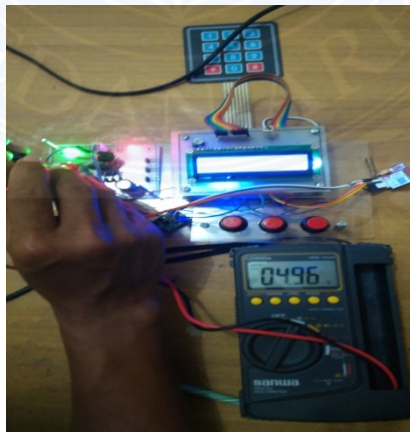
Perancangan program keseluruhan merupakan perancangan program mulai dari proses pembacaan sensor aktif alarm pada BTS hingga pada proses koneksi antara *mikrokontroler ATmega16* dengan modem *GSM* agar dapat melakukan proses kirim dan terima SMS dengan *handphone* tujuan.

Pengukuran dan implementasi ini meliputi, yaitu :

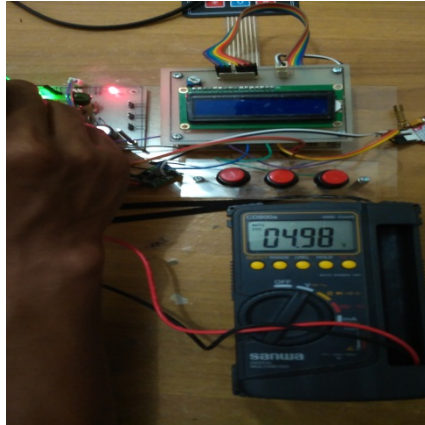
- 1) Pengukuran Rangkaian Sumber Tegangan DC 5V
- 2) Implementasi Tampilan LCD 16x2 Karakter.
- 3) Implementasi Kirim Dan Terima SMS Dengan Modem GSM.
- 4) Implementasi Keseluruhan

4.1. Pengukuran Rangkaian Sumber Tegangan DC 5V

Pengujian rangkaian sumber tegangan DC 5V dilakukan untuk mengetahui apakah rangkaian sumber tegangan DC 5V sudah dapat menghasilkan tegangan sebesar 5V yang digunakan untuk mencatu sistem keseluruhan. Hasil pengujian keluaran rangkaian sumber tegangan DC 5V ditunjukkan dalam Gambar 4.1 dan 4.2



Gambar 4.1 Hasil Keluaran Rangkaian Tanpa Beban



Gambar 4.2. Hasil Keluaran Rangkaian Berbeban

Berdasarkan hasil pengujian rangkaian sumber tegangan 5V dapat diketahui bahwa rangkaian sumber tegangan sebesar 4,96V pada keadaan tanpa beban dan menghasilkan tegangan sebesar 4,98V pada keadaan berbeban .

4.2. Implementasi LCD 16X2 Karakter

Implementasi LCD 16X2 karakter bertujuan untuk mengetahui kesesuaian antara karakter-karakter yang dikirimkan oleh program di dalam *mikrokontroler ATmega16* ke LCD dengan karakter yang tertampil pada layar LCD 16X2 karakter. Hasil pengujian LCD 16X2 karakter ditunjukkan dalam Gambar 4.3

Program LCD pada *mikrokontroler ATmega16* sebagai berikut:

```
#include <mega16.h>
// Alphanumeric LCD functions
#include <alcd.h>
#include <delay.h>
// Declare your global variables here
void main(void)
{
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
```



```

// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Imam Siregar");
delay_ms(1000);
while (1)
{
    // Place your code here
}
}

```



Gambar 4.3. Tampilan Hasil Rangkaian LCD 16x2

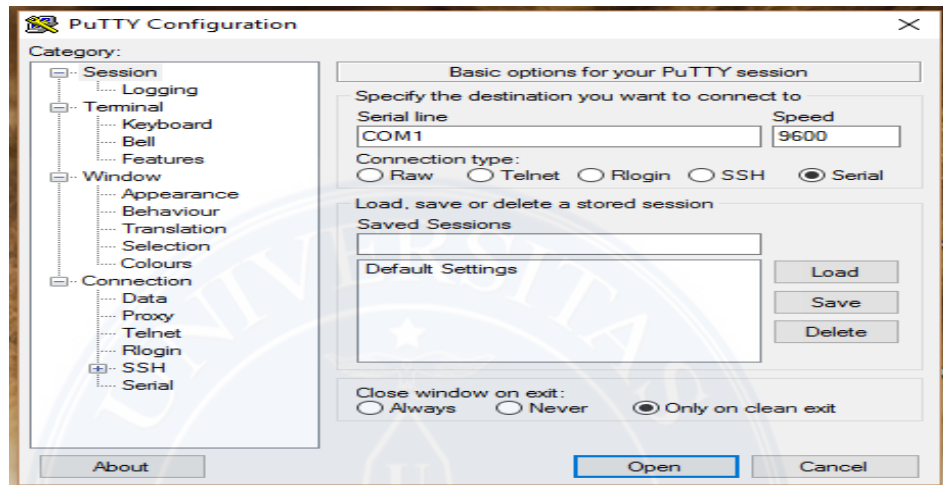
Hasil implementasi LCD 16x2 karakter menunjukkan bahwa *string* yang tertampil tiap baris LCD 16x2 karakter memiliki komposisi karakter yang identik dengan *string* yang dikirimkan mikrokontroler ATmega16 sehingga dapat disimpulkan bahwa LCD 16x2 karakter dapat berfungsi dengan baik.

4.3. Implementasi kirim dan terima SMS menggunakan modem GSM

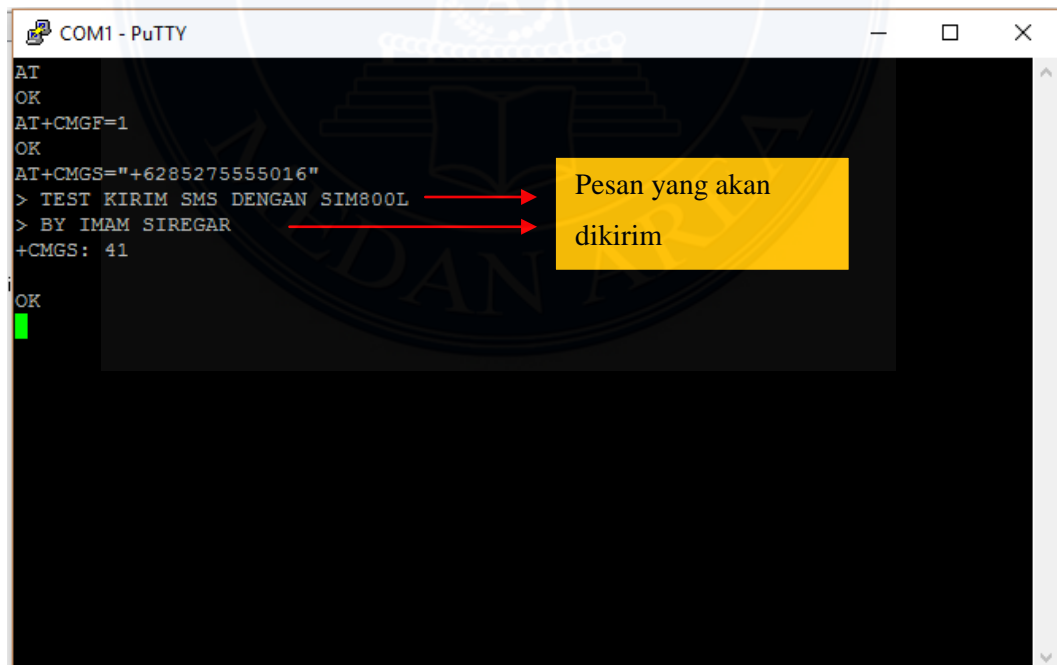
Implementasi kirim dan terima SMS menggunakan modem *GSM (General service for mobile)* dilakukan untuk mengetahui apakah modem GSM dapat mengirim dan menerima SMS dengan baik. Pengujian kirim dan terima SMS disimulasikan dengan menggunakan *Putty software*.

Putty software adalah *software remote console/ terminal* yang digunakan untuk *meremote* komputer dengan terhubungnya menggunakan port ssh atau sebagainya.

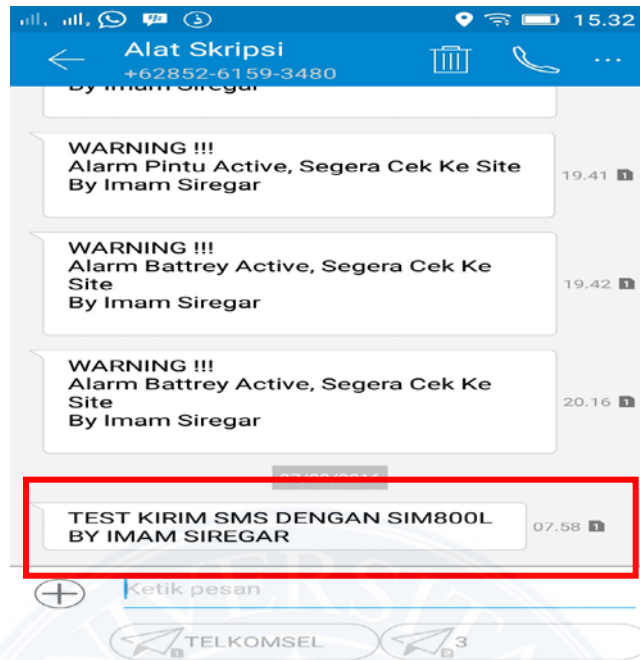
Hasil implementasi pengiriman SMS dari *Module GSM SIM800L* ditunjukkan dalam gambar berikut ini



Gambar 4.4 *Putty Configuration*

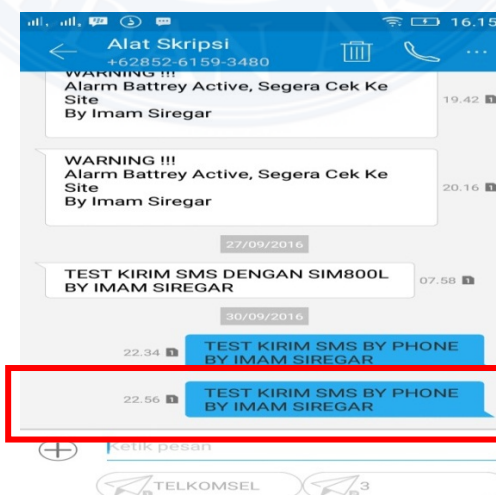


Gambar 4.5 Tampilan Pengujian Kirim SMS Pada *Software Putty*



Gambar 4.6 Tampilan SMS Yang Diterima Handphone

Berdasarkan hasil implementasi kirim SMS menggunakan *module GSM SIM 800L* dapat diketahui bahwa *module GSM* sukses mengirimkan SMS kepada handphone dengan isi pesan yang saling berkesesuaian antara pesan yang dikirim oleh *module GSM* dengan pesan yang diterima perangkat *handphone*. Sedangkan hasil implementasi terima SMS menggunakan *module GSM* ditunjukkan dalam gambar berikut



Gambar 4.7 Tampilan SMS Yang Dikirimkan Oleh *Handphone* Kepada *Module GSM*



Gambar 4.8 Tampilan implementasi terima SMS pada *software Putty*

Hasil implementasi terima SMS menggunakan modem GSM sukses menerima SMS dari perangkat *handphone* dengan isi pesan SMS yang saling berkesesuaian antara pesan yang diterima oleh modem GSM dengan pesan yang dikirim oleh perangkat *handphone*.

4.4. Implementasi Keseluruhan

Pengujian keseluruhan yang dilakukan terbagi menjadi 3 tahap yaitu pengujian aktif *alarm* pada pintu *shelter*, pengujian aktif alarm pada *battery* dan pengujian aktif alarm pada *feeder*.

4.4.1 Implementasi Aktif Alarm Pada Pintu Shelter

Pada pengujian ini list program sistem adalah sebagai berikut

```
void baca_sensor(){
    if ( door == on && batt == off && feeder == off)
    {
        sms_batt=0; sms_feeder=0;
        lcd_gotoxy(0,1);
        lcd_putsf("alarm pintu.");
        if ( sms_door == 0){
            send_sms();
        }
    }
}
```

```

printf("WARNING !!!\r\n");
printf("Alarm Pintu Active, Segera Cek Ke Site\r\n");
printf("By Imam Siregar\r\n");
delay_ms(1000);
putchar(26);
while (getchar()!='K'){};
lcd_gotoxy(0,1);
lcd_putsf("SMS SUKSES");
sms_door=1;
}
}

```



Gambar 4.9 Tampilan LCD Ketika Alarm Pintu Shelter Aktif

Berdasarkan hasil pengujian listing program pintu shelter, tampilan LCD dengan listing program saling berkesesuaian. Dimana pada alarm aktif ini mengartikan adanya seseorang yang masuk dalam shelter tanpa sepengetahuan user. Sehingga muncul kecurigaan tentang kondisi site (tower) tersebut secara real time. Atas kegagalan tersebut user mengunjungi site tersebut guna memastikan alarm aktif tersebut apakah berindikasi dengan stolen ataupun vandalism dll.

4.4.2 Implementasi Aktif Alarm Pada *Battery*

Pada pengujian ini list program sistem adalah sebagai berikut

```
if ( door == off && batt == on && feeder == off)
{
    sms_door=0; sms_feeder=0;
    lcd_gotoxy(0,1);
    lcd_putsf("alarm batt..");
    if ( sms_batt == 0){
        send_sms();
        printf("WARNING !!!\r\n");
        printf("Alarm Battrey Active, Segera Cek Ke Site\r\n");
        printf("By Imam Siregar\r\n");
        delay_ms(1000);
        putchar(26);
        while (getchar()!='K'){ };
        lcd_gotoxy(0,1);
        lcd_putsf("SMS SUKSES");
        sms_batt=1;
    }
}
```



Gambar 4.10 Tampilan LCD Ketika Alarm Baterai Aktif

Berdasarkan hasil pengujian listing program *battery*, tampilan LCD dengan *listing* program saling berkesesuaian. Pada alarm aktif ini mengartikan adanya seseorang ataupun makhluk hidup yang mencoba tidak memposisikan *battery* pada tempat semulanya lagi tanpa sepengetahuan user. Sehingga muncul kecurigaan tentang kondisi site (*tower*) tersebut secara *real time*. Oleh karena alarm tersebut *user* mengunjungi site tersebut guna memastikan alarm aktif tersebut apakah berindikasi dengan *stolen* ataupun *vandalism* dll.

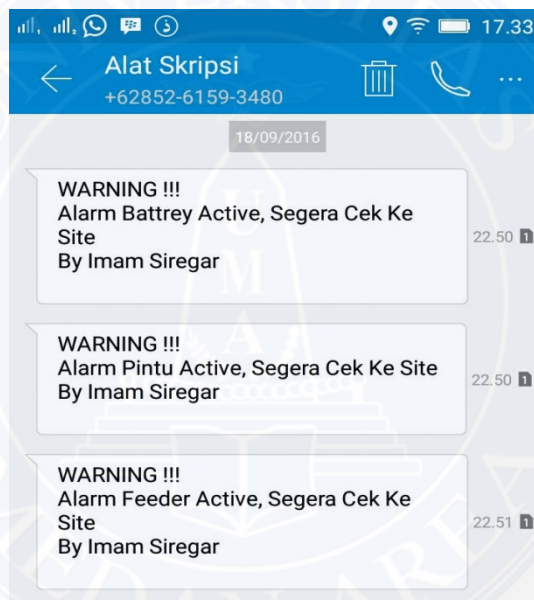
4.4.3 Implementasi Aktif Alarm Pada *Feeder*

Pada pengujian ini *list* program sistem adalah sebagai berikut

```
if ( door == off && batt == off && feeder == on)
{
    sms_door = 0; sms_batt=0;
    lcd_gotoxy(0,1);
    lcd_putsf("alarm feeder");
    if ( sms_feeder == 0){
        send_sms();
        printf("WARNING !!!\r\n");
        printf("Alarm Feeder Active, Segera Cek Ke Site\r\n");
        printf("By Imam Siregar\r\n");
        delay_ms(1000);
        putchar(26);
        while (getchar()!='K'){};
        lcd_gotoxy(0,1);
        lcd_putsf("SMS SUKSES");
        sms_feeder=1;
    }
}
```




Gambar 4.11 Tampilan LCD Ketika Alarm *Feeder* Aktif



Gambar 4.12 Tampilan SMS Aktif Alarm Pada Handphone

Dari pemberitahuan SMS aktif alarm pada gambar 4.12, *User* mengetahui bahwa adanya kondisi yang *abnormal* pada site (*tower*). Pemberitahuan aktif alarm tersebut pengguna HP dalam hal ini *team* cluster dilapangan harus memastikan lagi aktif alarm tersebut dengan mengunjungi site untuk memastikan site tersebut benar-benar aman atau ada kemungkinan indikasi kecurian pada perangkat *tower*. Dalam hal ini deteksi dini keamanan tower berhasil dilakukan sistem. Kecepatan respon *user* terhadap aktif alarm diperlukan gunaantisipasi terhadap hilangnya perangkat *tower*.