

## **TUGAS AKHIR**

### **RANCANG BANGUN *SMART ALARM* UNTUK MENGATASI GANGGUAN KEAMANAN PADA WILAYAH PERKOTAAN BERBASIS TEKNOLOGI SELULER DAN KOMPUTER**

Diajukan Sebagai Salah Satu Syarat Untuk Melaksanakan Tugas Akhir  
Pada Jurusan Teknik Elektro Fakultas Teknologi Industri  
Universitas Islam Indonesia



**Disusun Oleh :**

**Nama : Saiful Bahri Usman**

**No Mhs : 03524001**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2010**

## LEMBAR PENGESAHAN PEMBIMBING

RANCANG BANGUN *SMART ALARM* UNTUK MENGATASI GANGGUAN  
KEAMANAN PADA WILAYAH PERKOTAAN  
BERBASIS TEKNOLOGI SELULER DAN KOMPUTER



Pembimbing I

Tito Yuwono, ST.,M.Sc.

Pembimbing II

Medilla K, ST, M.Eng

## LEMBAR PENGESAHAN PENGUJI

**RANCANG BANGUN *SMART ALARM* UNTUK MENGATASI GANGGUAN  
KEAMANAN PADA WILAYAH PERKOTAAN  
BERBASIS TEKNOLOGI SELULER DAN KOMPUTER  
TUGAS AKHIR**

Oleh :  
Nama : SAIFUL BAHRI USMAN  
No. Mahasiswa : 03 524 001

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana pada Jurusan Teknik Elektro  
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Agustus 2011

Tim Penguji

Tit@ Yuwono, ST., M.Sc.  
Ketua

Wahyudi Budi Pramono, ST., M.Eng.  
Anggota I

Ir. Hj. Budi Astuti, MT.  
Anggota II

Mengetahui,

Ketua Jurusan Teknik Elektro  
Fakultas Teknologi Industri Universitas Islam Indonesia

Tito Yuwono, ST., M.Sc.

## KATA PENGANTAR



*Assalamu'alaikum Wr. Wb.*

Puji syukur yang sebesar-besarnya penulis haturkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga laporan tugas akhir dengan judul **“RANCANG BANGUN SMART ALARM UNTUK MENGATASI GANGGUAN KEAMANAN PADA WILAYAH PERKOTAAN BERBASIS TEKNOLOGI SELULER DAN KOMPUTER”** dapat terselesaikan. Laporan tugas akhir ini disusun sebagai syarat untuk memperoleh gelar Sarjana Teknik Elektro Universitas Islam Indonesia.

Penulis menyadari bahwa selama penulisan laporan tugas akhir ini tidak lepas dari bantuan, doa dan bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan kali ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. ALLAH SWT, yang selalu ada dalam setiap langkah dan dengan izin dan kuasa-Nya selalu memberi kesempatan dan kemudahan untuk dapat menyelesaikan tugas akhir ini.
2. Nabi Muhammad SAW, Nabi akhir zaman serta suri tauladan bagi seluruh umat-Nya.
3. Ayahanda H. Wahab Usman dan Ibunda Hj. Widiowati. yang senantiasa memberikan dukungan semangat, moril, materil dan do'a setiap saat.
4. Bapak Tito Yuwono, ST,M.Sc, selaku dosen pembimbing I tugas akhir yang telah memberikan bimbingan, masukan dan arahan dalam pembuatan alat dan penyusunan laporan tugas akhir.

5. Bapak Medilla K, ST, M.Eng selaku dosen pembimbing II tugas akhir yang telah memberikan bimbingan, masukan dan arahan dalam pembuatan alat dan penyusunan laporan tugas akhir.
6. Alm. Kakakku Yan Adriansyah Usman Terima kasih telah mendukung tiada henti-hentinya.
7. Segenap Dosen Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia yang telah membimbing dan memberikan sebagian ilmunya selama duduk di bangku kuliah.
8. Teman-temanku yang selalu mendukung dan membantuku dalam susah dan senang Helmi, Sigit, Agus, serta semua teman-teman TE UII khususnya angkatan 2003.
9. Untuk Yoannanda terima kasih untuk doa serta dukunganmu akhirnya tugas akhir ini dapat terselesaikan.
10. Serta semua pihak yang telah membantu, yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan yang ada pada laporan tugas akhir ini. Sehingga penulis sangat mengharapkan saran dan kritik yang membangun.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, Agustus 2011

**Saiful Bahri Usman**

*Halaman Motto*

--

*“Jadikanlah sabar dan shalat penolongmu dan sesungguhnya yang demikian itu sesungguhnya berat, kecuali bagi orang-orang yang khusyu”*

*(QS. Al Baqarah : 45)*

*“Do the best, God takes the rest”*

*(Dale Carnegie dalam buku “Positive thinking”)*

*”Perjalanan seribu batu bermula dari satu langkah”*

*(Lao Tze)*

*Detik demi detik terasa sekali. Sakit segala sakit mempunyai arti. Jadikanlah aku ya Allah sebagai ahli syukur karena masih dapat Kau uji. Apapun yang terjadi batasnya diketahui, bila tiba saatnya nanti, pasti akan terjadi. Nikmat yang kau berikan sungguh berarti, Subhannallah.*

## HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Tugas akhir ini Ku persembahkan untuk:*

*Ayah, dan ibuku Tercinta  
Langkah kaki & ketegaran dalam mendidik dan membimbingku  
bukanlah hitungan hari, minggu atau bulan bahkan tak terhitung  
tahun lagi, mampu diraih dengan senyuman & kesabaran.  
Tak tersirat pedih di wajah kalian meski berat beban di punggung  
kalian berjuang demi kebahagiaan Anakmu. Kesabaran dan  
ketabahan yang telah kalian tunjukkan telah mendewasakanmu.  
Inilah karya kecilku kupersembahkan untuk  
.....kalian berdua.....*

*Alm. Kakakku, Yan Adriansyah Usman  
(sigit, hekmy, agus dan yoannanda)  
Terima kasih atas inspirasinya dan selalu mendukungku!!!*

*Sahabat setia di semua penjuru langit  
Yang selalu membantuku dikala suka maupun duka dan memberikan  
ide-ide baru.*

*Karya ini juga ku persembahkan untuk diriku sendiri yang telah  
.....berusaha*

## DAFTAR ISI

|   | Halaman  |
|---|----------|
| Halaman Judul .....                               | i        |
| Pernyataan.....                                   | ii       |
| Halaman Pengesahan.....                           | iii      |
| Kata Pengantar.....                               | iv       |
| Intisari.....                                     | vi       |
| Daftar Isi.....                                   | vii      |
| Daftar Gambar.....                                | x        |
| Daftar Tabel.....                                 | xii      |
| <b>BAB I      PENDAHULUAN .....</b>               | <b>1</b> |
| 1.1    Latar Belakang Masalah .....               | 1        |
| 1.2    Rumusan Masalah .....                      | 2        |
| 1.3    Metodologi.....                            | 3        |
| 1.4    Manfaat Penelitian.....                    | 3        |
| 1.5    Tujuan Penelitian.....                     | 3        |
| 1.6    Sistematika Penulisan.....                 | 4        |
| <b>BAB II     LANDASAN TEORI.....</b>             | <b>5</b> |
| 2.1    Tinjauan Pustaka.....                      | 5        |
| 2.2    Mikrokontroler .....                       | 5        |
| 2.2.1.  Arsitektur Mikrokontroler ATmega8535..... | 7        |
| 2.2.2.  Fitur ATmega8535.....                     | 8        |
| 2.2.3.  Konfigurasi Pin ATmega8535.....           | 9        |
| 2.2.4.  Peta Memori.....                          | 10       |
| 2.3.    Telemetry.....                            | 11       |
| 2.4.    Short Message Service (SMS).....          | 12       |
| 2.4.1. PDU (Protocol Data Unit).....              | 13       |
| 2.4.2. SMS Deliver.....                           | 13       |



|         |   |    |
|---------|---|----|
|         | 2.4.3. SMS <i>Submit</i> .....                                    | 14 |
|         | 2.5 Komunikasi RS232.....   | 16 |
|         | 2.6 Visual Basic.....   | 16 |
| BAB III | METODOLOGI PENELITIAN.....  | 19 |
|         | 3.1 Perancangan Alat.....   | 19 |
|         | 3.2. Perancangan Perangkat Keras.....                             | 20 |
|         | 3.2.1. Alat dan Pengirim Data.....                                | 21 |
|         | 3.2.1.1. Switch.....  | 21 |
|         | 3.2.1.2. Mikrokontroler.....                                      | 21 |
|         | 3.2.1.3. Catu Daya.....   | 22 |
|         | 3.2.2. Piranti Keras Penerima Data dan Antarmuka<br>Komputer..... | 23 |
|         | 3.2.2.1. Mikrokontroler.....                                      | 23 |
|         | 3.2.1.2. Komunikasi Serial RS232.....                             | 24 |
|         | 3.3 Perancangan Perangkat Lunak.....                              | 25 |
|         | 3.3.1. Perancangan Piranti Lunak Mikrokontroler.....              | 25 |
|         | 3.3.1.1. Piranti Lunak Pengirim Data.....                         | 26 |
|         | 3.3.1.2. Piranti Lunak Antarmuka Komputer....                     | 27 |
|         | 3.3.2. Perancangan Piranti Lunak Komputer.....                    | 28 |
| BAB IV  | ANALISIS, DAN PEMBAHASAN.....                                     | 29 |
|         | 4.1 Hasil Perancangan Piranti Keras.....                          | 29 |
|         | 4.1.1. Rangkaian keseluruhan.....                                 | 28 |
|         | 4.1.2. Rangkaian Sensor.....                                      | 30 |
|         | 4.1.3. Rangkaian LCD .....  | 30 |
|         | 4.1.4. Catu Daya.....   | 31 |
|         | 4.2. Hasil Perancangan Piranti Lunak.....                         | 32 |
|         | 4.2.1. Inisialisasi awal.....                                     | 32 |
|         | 4.2.2. Inisialisasi LCD.....                                      | 33 |
|         | 4.3. Pengujian Keseluruhan.....                                   | 34 |

|       |                 |    |
|-------|-----------------|----|
| BAB V | PENUTUP.....    | 35 |
| 5.1   | Kesimpulan..... | 35 |
| 5.2   | Saran.....      | 35 |
|       | DAFTAR PUSTAKA  | 37 |
|       | LAMPIRAN        |    |



## DAFTAR GAMBAR

|             | Halaman  |
|-------------|--|
| Gambar 2.1  | Blok diagram fungsional ATmega8535..... 7                    |
| Gambar 2.2  | Pin ATmega8535..... 9  |
| Gambar 2.3  | Memori ATmega8535..... 11                                    |
| Gambar 2.4  | Sistem Telemetri..... 12                                     |
| Gambar 2.5  | Format PDU untuk SMS <i>deliver</i> ..... 14                 |
| Gambar 2.6  | Format PDU untuk SMS <i>submit</i> ..... 15                  |
| Gambar 2.7  | Rangkaian koneksi RS232 dengan mikrokontroler..... 16        |
| Gambar 3.1  | Diagram Blok Sistem Pemancar..... 19                         |
| Gambar 3.2  | Diagram Blok Sistem Penerima..... 19                         |
| Gambar 3.3  | Rangkaian switch..... 21                                     |
| Gambar 3.4  | Rangkaian Pengirim Data..... 22                              |
| Gambar 3.5  | Rangkaian Catu Daya..... 23                                  |
| Gambar 3.6  | Rangkaian Mikrokontroler Penerima Data..... 24               |
| Gambar 3.7  | Rangkaian komunikasi serial RS232..... 25                    |
| Gambar 3.8  | Diagram alir Piranti Lunak Pengirim Data Mikrokontroler.. 26 |
| Gambar 3.9  | Diagram alir Piranti Lunak Antarmuka Mikrokontroler..... 27  |
| Gambar 3.10 | Diagram alir Piranti Lunak Komputer..... 28                  |
| Gambar 4.1  | Rangkaian keseluruhan alat..... 29                           |
| Gambar 4.2  | Gambar Rangkaian LCD..... 31                                 |
| Gambar 4.3  | Catu Daya..... 31  |

## DAFTAR TABEL

|  | Halaman |
|--|---------|
| Tabel 3.1 Koneksi Port Mikrokontroler.....               | 22      |
| Tabel 3.2 Koneksi Port Mikrokontroler Penerima Data..... | 24      |
| Tabel 4.1 Pengukuran sensor 1 (optokopler).....          | 30      |
| Tabel 4.2 Pengukuran sensor 2 (switch).....              | 30      |
| Tabel 4.3 Pengujian Catu Daya.....                       | 32      |
| Tabel 4.4 Pengujian Fungsional Alat.....                 | 34      |



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Tingkat kejahatan dari hari ke hari menjadi semakin meningkat dengan adanya berbagai macam tindakan kriminal (perampokan/pencurian). Pada saat kejahatan terjadi Polisi/Staf Keamanan menjadi terlambat mencapai lokasi kejahatan karena laporan yang masuk datang terlambat. Terutama pada aksi perampokan yang menimpa pada suatu wilayah perkotaan terutama pada perkantoran di siang atau malam hari, kehadiran aparat keamanan dengan cepat dan tepat sangatlah diperlukan. Sistem komunikasi melalui telepon akan menjadi tidak bermanfaat manakala suatu tempat mengalami kejadian perampokan. Sistem keamanan suatu wilayah perkotaan pada umumnya adalah dengan mengandalkan satpam (petugas keamanan), sehingga apabila terjadi gangguan pada salah satu wilayah tertentu tidak dapat diketahui secara langsung oleh petugas keamanan yang berada di sentral keamanan. Perangkat keamanan dengan menggunakan sistem kabel juga sangat rawan terhadap terjadinya gangguan apabila kabel komunikasi diputus oleh pelaku kejahatan. Selain itu kelemahan dari sistem keamanan ini adalah bahwa pencurian baru bisa terdeteksi pada saat petugas melakukan pengecekan.

Berdasarkan hal itu maka muncul permasalahan bagaimana mewujudkan pemantau keamanan suatu wilayah perkotaan yang dapat segera memberi tanda kepada petugas di pos keamanan sentral manakala terjadi pencurian yang diawali dengan kerusakan, maka perlu dirancang suatu mekanisme *alarm* secara jarak

jauh yang dapat membuat pihak keamanan mengetahui bahwa telah atau tengah terjadi tindak kejahatan pada suatu tempat wilayah. Mekanismenya adalah : Pada kantor kepolisian atau bagian pengawasan suatu wilayah terdapat suatu piranti lunak yang akan menghadirkan lokasi kejahatan secara cepat dalam hitungan menit. Berbekal data lokasi inilah maka kepolisian atau satuan keamanan akan secara cepat melakukan penanganan aktivitas kejahatan.

Sistem dikembangkan dengan menggunakan jaringan GSM yang telah ada, yaitu menggunakan layanan SMS dan *Call* dari operator, namun untuk mengatasi kendala yang mungkin terjadi pada saat jaringan SMS sedang sibuk dapat pula digunakan layanan *call* atau *voice*. Program yang dibuat pada komputer dibangun menggunakan bahasa pemrograman *Visual Basic*.

## **1.2. Perumusan Masalah**

Permasalahan dalam penelitian yang akan dilakukan dapat dirumuskan dalam beberapa pertanyaan sebagai berikut :

1. Bagaimana membuat sebuah peralatan elektronik yang mampu mengatasi kesulitan pemantauan keamanan dan potensi apabila terjadi pencurian wilayah perkotaan ataupun suatu tempat yang mempunyai lokasi yang jaraknya berjauhan?
2. Bagaimana supaya pusat pengamanan Bank secara sentral dapat lebih cepat mengidentifikasi adanya pencurian di wilayah pengawasan sekaligus lokasi kejadian dengan menggunakan jaringan yang ada seperti jaringan GSM ?

### 1.3. Batasan Masalah

Masalah yang akan dibahas adalah sistem *smart alarm* untuk mengatasi pencurian pada wilayah perkotaan berbasis teknologi seluler dan komputer, untuk itu pada penelitian ini masalah yang dibatasi adalah:

- a. Perancangan yang dilakukan adalah piranti keras dan piranti lunak antara lain: personal komputer, mikrokontroler dan pemrograman *visual basic*.
- b. Pada tampilan di komputer menggunakan dua tempat/wilayah.

### 1.4. Manfaat Penelitian

Manfaat dari penelitian ini adalah agar mampu mengembangkan sistem perangkat *smart alarm* untuk mengatasi pencurian/kejahatan di suatu wilayah. Dengan menggunakan sistem teknologi seluler dan komputer maka kejadian pencurian/kejahatan dapat diatasi lebih cepat.

### 1.5. Tujuan Penelitian

Berdasarkan latar belakang masalah di atas, maka tujuan dari penelitian ini adalah merancang sistem smart alarm untuk mengetahui kejahatan yang terjadi pada perkotaan secara akurat sehingga dapat diatasi dengan cara yang cepat oleh pihak yang berwajib.

## **1.6. Sistematika Penulisan**

### **BAB I. PENDAHULUAN**

Dalam bab ini penulis mengemukakan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

### **BAB II. LANDASAN TEORI**

Dalam bab ini, penulis mengemukakan teori-teori tentang mikrokontroler ATmega8535, telemetri, SMS, komunikasi RS 232, dan pemrograman *visual basic* yang berhubungan dengan masalah yang akan dibahas dalam tugas akhir ini. Hal ini sangat penting, sebab merupakan landasan dalam pemecahan masalah dengan teori-teori tersebut.

### **BAB III. METODOLOGI PENELITIAN**

Dalam bab ini, penulis menyajikan obyek penelitian, pengumpulan data, kerangka penelitian, dan menganalisa permasalahan. .

### **BAB IV. ANALISIS DAN PEMBAHASAN**

Dalam bab ini, penulis menyajikan tentang teknik analisa yang dilakukan sehingga dicapai hasil analisis..

### **BAB V. PENUTUP**

Dalam bab ini, penulis menyajikan tentang kesimpulan dan saran-saran terhadap tugas akhir ini.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Sistem pengamanan *Smart Alarm* memang membuat *handphone* layaknya satpam. Seperangkat alat itu dipasangkan dalam mobil, lalu digabung dengan *SIM card phonesel*. Jadi, mobil tersebut punya nomor *handphone* sendiri. Perangkat itu lalu dihubungkan dengan alarm mobil dan sistem kelistrikan dalam kendaraan tersebut. *Smart Alarm* dirancang dan dirakit oleh Ir. Safril, dosen Politeknik ITB. Sekitar Agustus tahun 2002, penelitian tersebut dibiayai oleh Afiscom.

Dalam penelitian ini akan menggunakan pendekatan yang berbeda dengan penelitian terdahulu yaitu dengan menggunakan mikrokontroler ATmega8535, sistem yang dibangun menggunakan basis SMS dengan terkoneksi ke komputer.

#### **2.2. Mikrokontroler**

Mikrokontroler adalah suatu terobosan teknologi mikroprosesor dan mikrokomputer, yang mana teknologi ini adalah teknologi semikonduktor dengan kandungan transistor yang lebih banyak, namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara masal (dalam jumlah banyak) sehingga harganya menjadi lebih murah.

Mikrokontroler ini kemampuan digitalnya menirukan fungsi otak manusia, sehingga meliputi fungsi atau instruksi aritmatika (berhitung), logika (mempertimbangkan suatu kondisi), dan memori. Mikrokontroler ini berbeda

halnya dengan mikroprosesor yang hanya pemrosesannya terdiri dari *Central Processing Unit* (CPU) dan register-register, tanpa memori, tanpa I/O, dan *peripheral* yang dibutuhkan oleh suatu sistem supaya dapat bekerja.

Mikrokontroler adalah sebuah mikrokontroler yang terdiri dari 8 bit dengan *low power* dan performa tinggi. Termasuk dalam mikrokontroler AVR yang memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*. .

Mikrokontroler AVR memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O dan 512 *byte* SRAM *Internal*.

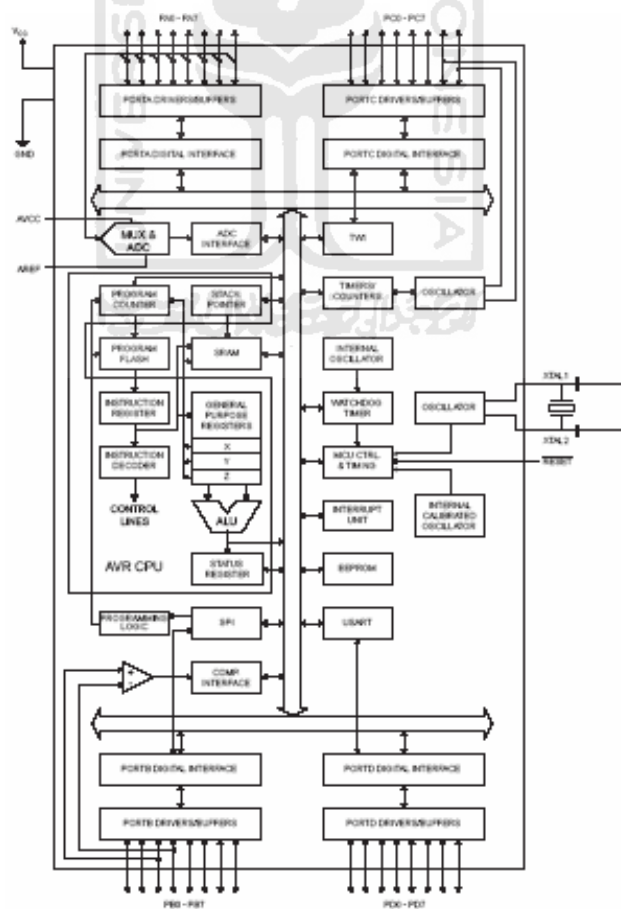
Register keperluan umum menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikut, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* mikrokontroler, seperti kontrol register, *timer/counter*, fungsi-fungsi I/O dan sebagainya.

Memori program yang terletak dalam *Flash* PEROM tersusun dalam *word* atau 2 *byte* karena setiap instruksi memiliki lebar 16-bit atau 32-bit. AVR memiliki 4K *Byte* X16-bit *Flash* PEROM dengan alamat mulai dari \$000 sampai \$FFF. AVR tersebut memiliki 12-bit *Program Counter* (PC) sehingga mampu mengalami isi *Flash*.

Selain itu mikrokontroler AVR juga memiliki memori data berupa EEPROM 8-bit sebanyak 512 *byte*. Alamat EEPROM dimulai dari \$000 sampai \$1FF.

### 2.2.1. Arsitektur Mikrokontroler AVR ATmega8535

Mikrokontroler ATmega8535 adalah sebuah mikrokontroler yang terdiri dari 8 bit dengan *low power* dan performa tinggi. Termasuk dalam mikrokontroler AVR yang memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*.



Gambar 2.1 Blok diagram fungsional ATmega8535

Dari gambar tersebut dapat dilihat bahwa ATmega8535 memiliki bagian sebagai berikut:

1. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C* dan *Port D*.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog Timer* dengan isolator internal.
6. SRAM sebesar 512 *byte*.
7. Memori *Flash* sebesar 8 kb dengan kemampuan *Read While Write*.
8. unit interupsi internal dan eksternal.
9. *Port* antarmuka SPI.
10. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. *Port* USART untuk komunikasi serial.

### 2.2.2 Fitur ATmega8535

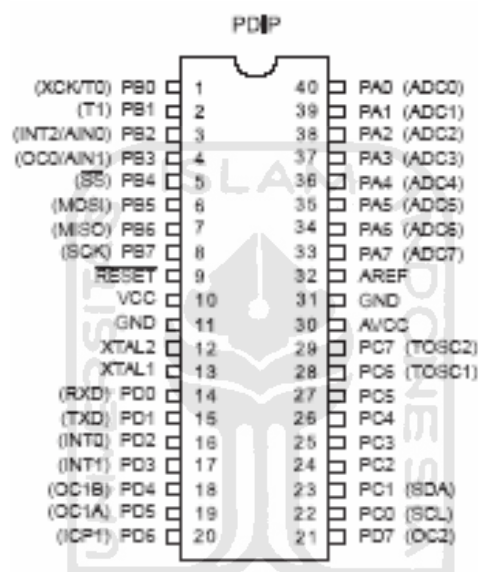
Kapabilitas detail dari ATmega8535 adalah sebagai berikut:

1. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
2. Kapabilitas memori *flash* 8 KB, SRAM sebesar 512 *byte*, dan EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 *byte*.
3. ADC internal dengan fidelitas 10 bit sebanyak 8 *channel*.

4. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* menghemat penggunaan daya listrik.

### 2.2.3 Konfigurasi Pin ATmega8535

Konfigurasi pin ATmega8535 bisa dilihat pada Gambar 2.2 berikut ini.



Gambar 2.2 Pin ATmega8535

Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega8535 sebagai berikut:

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground*.
3. *Port A* (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. *Port B* (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog dan SPI

5. *Port C* (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan *Timer Oscillator*.
6. *Port D* (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

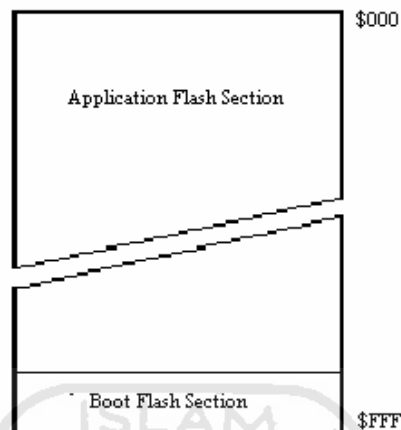
#### 2.2.4 Peta Memori

AVR ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O dan 512 *byte* SRAM *Internal*.

Register keperluan umum menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikut, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler, seperti kontrol register, *timer/counter*, fungsi-fungsi I/O dan sebagainya.

Memori program yang terletak dalam *Flash* PEROM tersusun dalam *word* atau 2 *byte* karena setiap instruksi memiliki lebar 16-bit atau 32-bit. AVR ATmega8535 memiliki 4K *Byte* X16-bit *Flash* PEROM dengan alamat mulai dari

\$000 sampai \$FFF. AVR tersebut memiliki 12-bit *Program Counter* (PC) sehingga mampu mengamati isi *Flash*.



Gambar 2.3 Memori ATmega8535

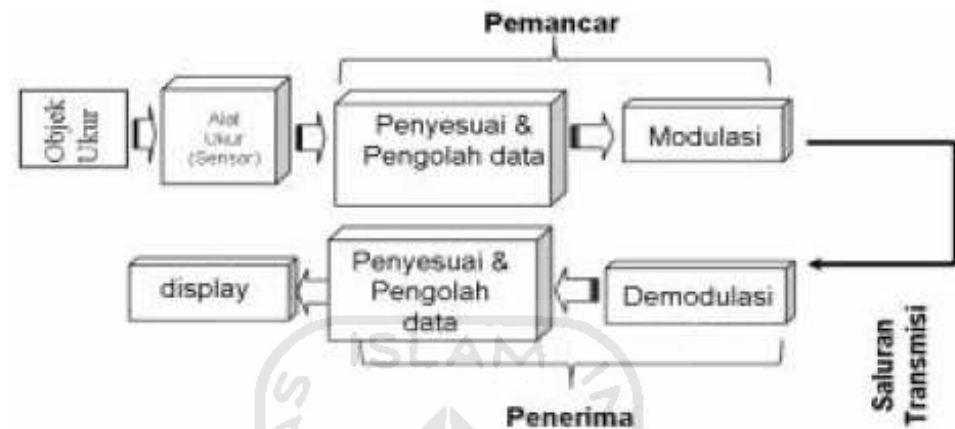
Selain itu, AVR ATmega8535 juga memiliki memori data berupa EEPROM 8-bit sebanyak 512 *byte*. Alamat EEPROM dimulai dari \$000 sampai \$1FF.

### 2.3. Telemetri

Kata telemetri berasal dari akar bahasa Yunani *tele* = jarak jauh, dan *metron* = pengukuran. Telemetri adalah sebuah teknologi yang melakukan pengukuran jarak jauh dan pelaporan informasi kepada perancang atau operator sistem.

Telemetri adalah proses pengukuran parameter suatu obyek (benda, ruang, kondisi alam), yang hasil pengukurannya di kirimkan ke tempat lain melalui proses pengiriman data baik dengan menggunakan kabel maupun tanpa menggunakan kabel (*wireless*), selanjutnya data tersebut untuk dimanfaatkan

langsung atau perlu dianalisa. Secara umum sistem telemetri terdiri atas enam bagian pendukung yaitu objek ukur, sensor, pemancar, saluran transmisi, penerima dan tampilan/display.



Gambar 2.4. Sistem Telemetri

#### 2.4. Short Message Service (SMS)

*Short Message Service* (SMS) merupakan fasilitas pengiriman pesan singkat antar telepon seluler yang terdapat khusus pada telepon seluler digital. Dikatakan pesan singkat karena pengiriman pesannya dibatasi dalam beberapa karakter tertentu. SMS dikirim dari telepon seluler ke telepon seluler lain harus melewati *Short Message Service Center* (SMSC) yang akan menyimpan sementara pesan yang dikirim lalu meneruskannya pada penerima.

Apabila penerima pesan dalam keadaan tidak aktif atau berada di luar jangkauan operator maka SMSC akan menyimpan pesan sementara dan mengirimkan lagi apabila penerima sudah memungkinkan untuk menerima pesan.



Fasilitas SMS dapat dihubungkan langsung dengan perlengkapan terminal luar seperti PC untuk mengontrol SMS pada telepon seluler.

Koneksi antara perlengkapan terminal luar dengan SMS pada telepon seluler dapat diwujudkan dengan serial kabel, *bluetooth*, infra merah, dan lain-lain. Aturan-aturan antarmukanya berdasarkan pada *AT Commands*.

#### **2.4.1. PDU (*Protocol Data Unit*)**

Ada dua cara atau mode dalam pengiriman dan penerimaan pesan SMS yaitu dengan mode teks dan mode PDU. Dalam mode PDU pesan yang dikirim tidak hanya berisi teks tetapi juga berisi informasi-informasi penting lainnya seperti nomor SMSC (*Short Message Service Centre*), nomor pengirim atau penerima, validitas waktu, waktu pengiriman, dan lain-lain.

Cara penulisan dalam mode PDU ini adalah dalam bentuk pasangan-pasangan bilangan hexadesimal atau yang biasa disebut pasangan oktet. Metode perpindahan SMS menggunakan mode PDU dibedakan menjadi dua yaitu SMS *deliver* dan SMS *submit*.

#### **2.4.2. SMS *Deliver***

SMS *deliver* adalah pesan yang diterima oleh telepon seluler dari SMSC.

Ada delapan *header* SMS *deliver* yaitu:

a. *Service Centre Address* (SCA)

Berisi informasi SMS Centre.

b. *TPDU Type Identifier* (ID)

Berisi informasi jenis dari PDU tersebut.

c. *Originating Address* (OA)

Berisi informasi nomor pengirim.

d. *Protocol Identifier (PID)*

Berisi informasi identifikasi protokol yang digunakan.

e. *Data Coding Scheme (DCS)*

Berisi informasi skema pengkodean data yang digunakan.

f. *Service Centre Time Stamp (SCTS)*

Berisi informasi waktu.

g. *User Data Length (UDL)*

Berisi informasi panjang dari data yang dibawa.

h. *User Data (UD)*

Berisi informasi data utama yang dibawa.

PDU harus ditulis berdasarkan format yang telah ditetapkan. Format PDU untuk SMS *deliver* ditunjukkan pada gambar 2.5.



Gambar 2.5. Format PDU untuk SMS *deliver*

### 2.4.3. SMS *Submit*

SMS *Submit* adalah pesan yang dikirim oleh telepon seluler ke SMSC. Ada sembilan *header SMS submit* yaitu:

a. *Service Centre Address (SCA)*

Berisi informasi SMS Centre.

b. *TPDU Type Identifier (ID)*

Berisi informasi jenis dari PDU tersebut.

c. *Message Reference (MR)*

Parameter yang mengindikasikan nomor referensi SMS *Submit*.

d. *Destination Address (DA)*

Berisi informasi nomor alamat yang dituju.

e. *Protocol Identifier (PID)*

Berisi informasi identifikasi protokol yang digunakan.

f. *Data Coding Scheme (DCS)*

Berisi informasi skema pengkodean data yang digunakan.

g. *Validity Period (VP)*

Berisi jangka waktu validitas pesan pada jaringan.

h. *User Data Length (UDL)*

Berisi informasi panjang dari data yang dibawa.

i. *User Data (UD)*

Berisi informasi data utama yang dibawa.

Sembilan *header SMS submit* ditulis berdasarkan format PDU untuk SMS *submit*. Format PDU untuk SMS *submit* ditunjukkan pada gambar 2.6.

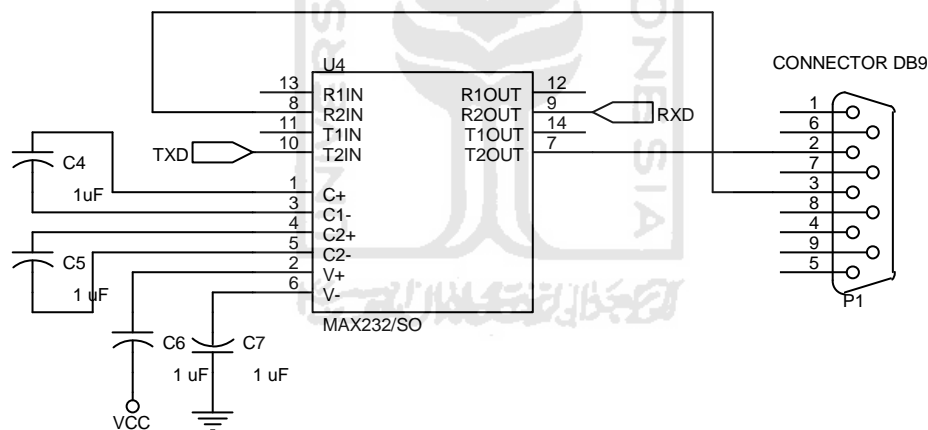
|                 |      |    |     |     |    |     |           |
|-----------------|------|----|-----|-----|----|-----|-----------|
| SCA             | IDMR | DA | PID | DCS | VP | UDL | USER DATA |
| SMS-SUBMIT TPDU |      |    |     |     |    |     |           |

Gambar 2.6. Format PDU untuk SMS *submit*

## 2.5. Komunikasi RS232

Komunikasi serial RS232 digunakan sebagai antarmuka antara komputer dengan mikrokontroler. Agar level tegangan data serial dari mikrokontroler setara dengan level tegangan komunikasi port serial PC, diperlukan MAX232 untuk mengubah ke tegangan TLL/CMOS *logic level* RS232. .

Ada tiga hal pokok yang diatur standard RS232, antara lain bentuk sinyal dan level tegangan yang dipakai, penentuan jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki-kaki di konektor, penentuan tata cara pertukaran informasi antara komputer dan alat-alat pelengkap. Gambar 2.7 berikut menunjukkan rangkaian koneksi RS232 dengan mikrokontroler.



Gambar 2.7. Rangkaian koneksi RS232 dengan mikrokontroler

## 2.6. Visual Basic

Untuk membaca data dari mikrokontroler dan menampilkannya pada komputer diperlukan suatu piranti lunak, piranti lunak yang digunakan adalah *Visual Basic*.

Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu (Kasmoni, 2005). *Visual Basic* adalah salah satu bahasa pemrograman komputer generasi ke-4 (*fourth generation programming language*) yang diciptakan oleh Bill Gates. Bahasa pemrograman *Visual Basic*, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Bill Gates, pendiri Microsoft, memulai bisnis *software*nya dengan mengembangkan *Basic Interpreter* untuk IBM. Perkembangan berikutnya ialah diluncurkannya BASICA (*basic-advanced*) untuk DOS. Setelah BASICA, Microsoft meluncurkan *Microsoft QuickBasic* dan *Microsoft Basic* (dikenal juga sebagai *Basic Compiler*).

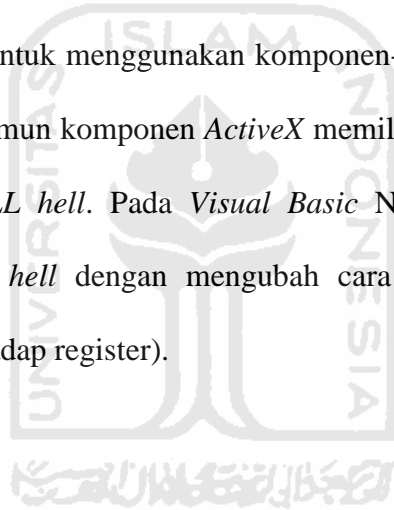
*Visual Basic* merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi *Windows*. *Visual Basic* merupakan salah satu bahasa pemrograman komputer yang mendukung *object* (*Object Oriented Programming /OOP*).

*Visual Basic* merupakan gabungan dua proyek *Microsoft* yaitu bahasa yang diinterpretasi (BASICA) dan bahasa yang dikompilasi (BASCUM) dan juga sebagai implementasi gabungan keduanya. *Programmer* yang menggunakan *Visual Basic* bisa memilih kode terkompilasi atau kode yang harus diinterpretasi sebagai hasil *executable* dari kode *Visual Basic*.

Kekurangan dari *Visual Basic* adalah meskipun sudah terkompilasi jadi bahasa mesin, DLL bernama MSVBVMxx.DLL tetap dibutuhkan. Namun

karakteristik bahasa terkompilasi tetap muncul (akan lebih cepat daripada seandainya kita menggunakan mode terinterpretasi). Meskipun *Visual Basic* termasuk bahasa pemrograman yang mendukung OOP, namun tidak sepenuhnya kompatibel. Beberapa karakteristik obyek tidak dapat dilakukan pada *Visual Basic*, seperti *Inheritance* tidak dapat dilakukan pada *class module*. *Polymorphism* secara terbatas bisa dilakukan dengan mendeklarasikan *class module* yang memiliki *Interface* tertentu.

*Visual Basic* menjadi populer karena kemudahan desain *form* secara visual dan adanya kemampuan untuk menggunakan komponen-komponen *ActiveX* yang dibuat oleh pihak lain. Namun komponen *ActiveX* memiliki masalahnya tersendiri yang dikenal sebagai *DLL hell*. Pada *Visual Basic* NET, *Microsoft* mencoba mengatasi masalah *DLL hell* dengan mengubah cara penggunaan komponen (menjadi independen terhadap register).



## BAB III

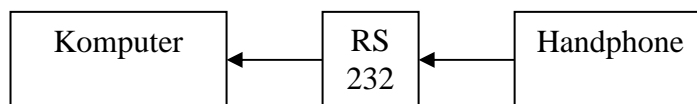
### METODOLOGI PENELITIAN

#### 3.1 Perancangan Alat

Alat yang akan dibuat sesuai dengan perumusan masalah yang ada menggunakan beberapa bagian utama yang saling bekerja sama membentuk sistem pemancar dan penerima. Sistem terdiri dari 2 bagian utama yaitu bagian pemancar yang berada di suatu tempat seperti pada meja pada toko emas atau bangunan bank dan bagian penerima yang berada di sentral kantor polisi atau pos keamanan sebagai operator. Berikut ini adalah diagram blok gambaran dari sistem secara keseluruhan.



Gambar 3.1. Diagram Blok Sistem Pemancar



Gambar 3.2. Diagram Blok Sistem Penerima

Cara kerja sistem adalah sebagai berikut :

1. Ketika terjadi kejahatan/perusakan, maka switch pada alat yang diletakkan di suatu tempat yang tidak kelihatan oleh pelaku kejahatan akan ditekan oleh korban. Apabila terjadi kejahatan switch ditekan, maka alat akan mengirimkan sms dan melakukan panggilan voice ke nomor telepon pada alat yang terpasang di kepolisian/Pos keamanan sentral.
2. Komputer utama yang diletakkan pada kantor polisi/pos keamanan sentral akan membunyikan *alarm* dan menunjukkan titik lokasi berlangsungnya kejahatan pada peta.
3. Data kemudian akan diterima oleh operator melalui *handphone* penerima dan antarmuka yang terhubung ke komputer. Data secara otomatis akan di-*log* oleh komputer dan dimasukkan ke file *database*.
4. Apabila data telah diterima, maka operator dapat memberikan informasi kepada petugas bank dan aparat kepolisian dapat langsung menuju TKP (Tempat Kejadian Perkara).

### **3.2. Perancangan Perangkat Keras**

Untuk dapat menjalankan sistem dibutuhkan piranti keras. Sesuai dengan konsep pada Bab 1 maka perangkat keras memiliki beberapa bagian utama piranti keras alat yang dipasang pada suatu tempat (meja) dan pengirim data dan *handphone*.

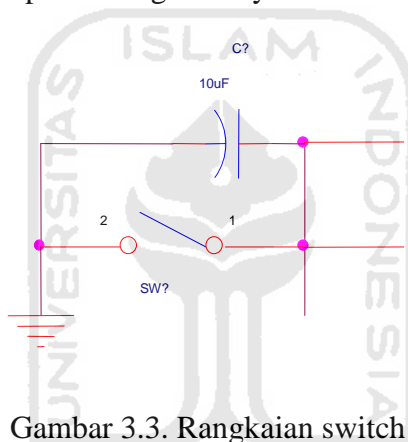


### 3.2.1. Alat dan Pengirim Data

Rangkaian alat dan pengirim data terdiri dari beberapa unit piranti keras yang mendukung sehingga terbentuk satu sistem pengiriman data. Berikut ini akan dijabarkan perancangan piranti keras pendukung tersebut.

#### 3.2.1.1. Switch

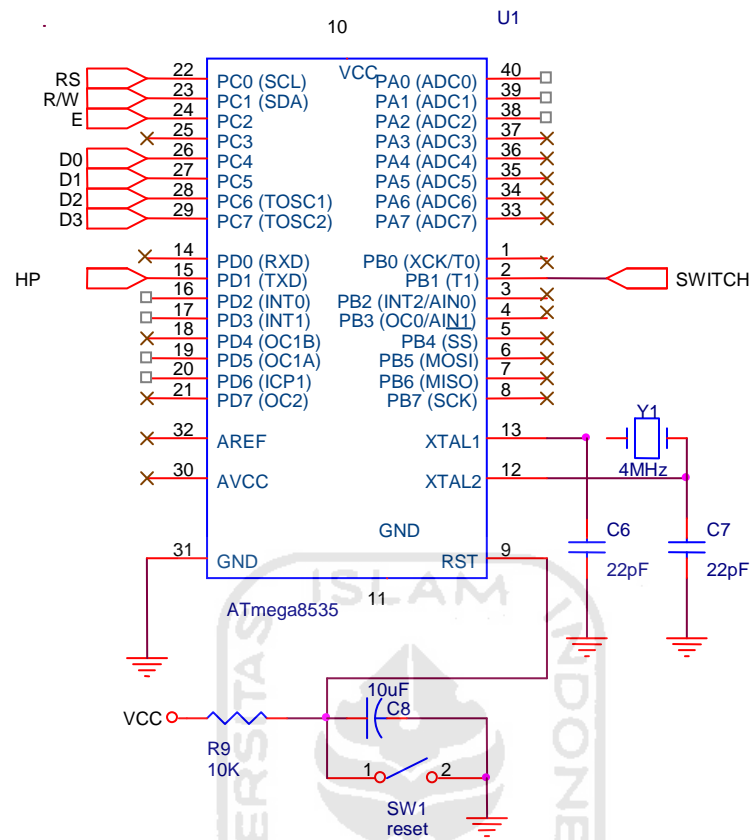
Switch yang ada digunakan untuk mendeteksi apabila terjadi kejahatan dan alat akan dijalankan secara simulasi. Dimana switch ini merupakan sensor yang ada pada alat, berikut merupakan rangkaiannya.



Gambar 3.3. Rangkaian switch

#### 3.2.1.2. Mikrokontroler

Switch yang telah dirancang membutuhkan suatu unit proses agar dapat beroperasi, oleh karena itu diperlukan suatu piranti mikrokontroler yang dapat mengontrol seluruh sistem dan melakukan proses konversi dan penghitungan data. Berikut ini gambar rancangan dari piranti mikrokontroler yang digunakan pada perancangan :



Gambar 3.4. Rangkaian Pengirim Data

Berikut ini tabel koneksi antara sensor dengan piranti lain

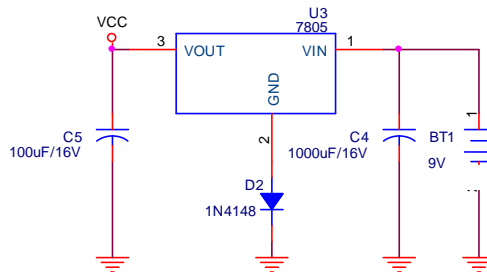
Tabel 3.1. Koneksi Port Mikrokontroler

| Port | Koneksi       | fungsi   |
|------|---------------|--|
| PD0  | Tx Handphone  | Mengirimkan data ke Penerima melalui handphone |
| PD1  | Rx Handphone  |  |
| PB1  | Switch Sensor | Panic Button                                   |
|      |               |  |

### 3.2.1.3. Catu Daya

Sebagai sumber daya dari rangkaian menggunakan sumber jala-jala PLN 220V, namun karena rangkaian menggunakan catuan 5V, maka perlu

menggunakan penurun tegangan. Berikut ini rangkaian yang digunakan sebagai catu daya :



Gambar 3.5. Rangkaian Catu Daya

Dari gambar di atas, beberapa fungsi dari rangkaian dapat diterangkan sebagai berikut:

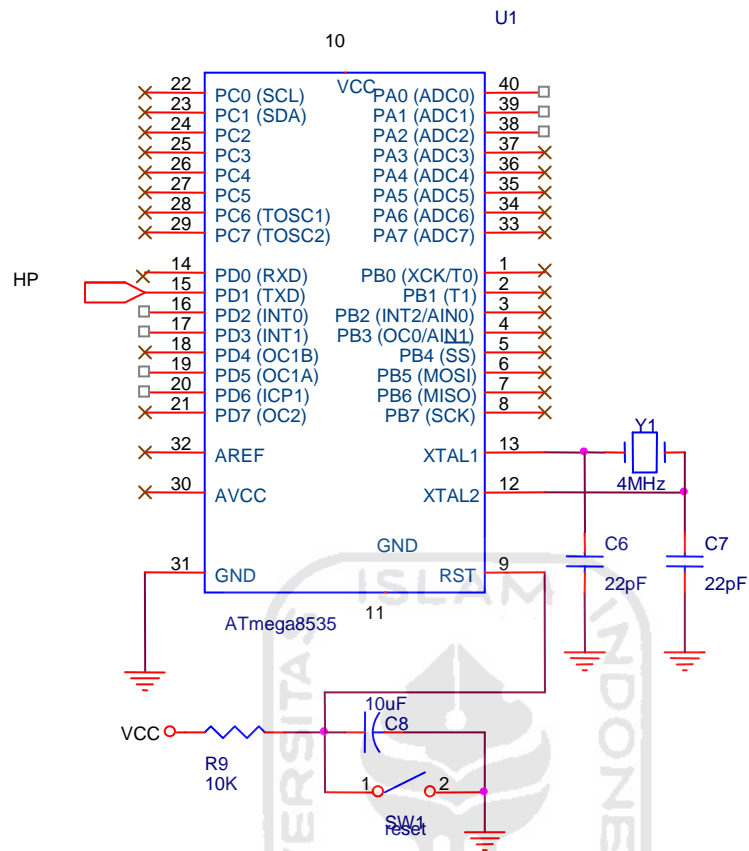
1. Kapasitor C4 sebagai penghilang *ripple* (filter) awal
2. IC LM7805 sebagai regulator penurun tegangan 9 menjadi 5 V
3. Dioda D1 berfungsi untuk menaikkan tegangan menjadi 5V
4. Kapasitor C5 sebagai filter dan stabilizer akhir

### 3.2.2. Piranti Keras Penerima Data dan Antarmuka Komputer

Rangkaian ini berfungsi untuk menerima data dari komputer dan meneruskannya ke komputer untuk diolah oleh *Visual Basic*. Pada bagian ini piranti keras terdiri dari mikrokontroler, RS 232, dan catu daya.

#### 3.2.2.1. Mikrokontroler

Mikrokontroler berfungsi untuk menerima data secara serial dari radio taxi kemudian akan meneruskannya ke komputer melalui converter RS 232. Berikut ini adalah perancangan dari piranti keras antarmuka komputer yang akan dibuat :



Gambar 3.6. Rangkaian Mikrokontroler Penerima Data

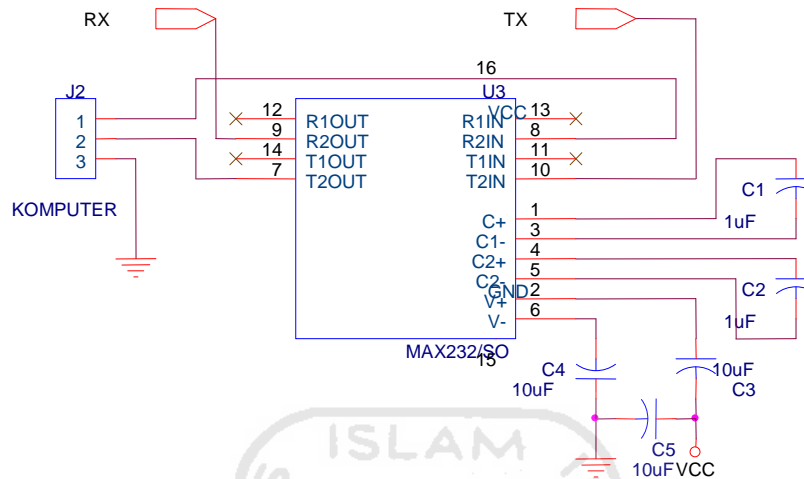
Tabel 3.2 Koneksi Port Mikrokontroler Penerima Data

| Port | Koneksi          | Fungsi   |
|------|------------------|--|
| PD0  | Tx FSK Modulator | Mengirimkan data ke Penerima melalui demodulator |
| PD1  | Rx FSK Modulator |  |
| PB3  | Switch           | Mengaktifkan Mode transmit                       |

### 3.2.2.2. Komunikasi Serial RS232

Pada sistem yang dibangun diperlukan adanya koneksi antara komputer dengan *handphone* supaya sistem berjalan. Dalam menjalankan sistem ini alat

koneksi tersebut adalah kabel data serial MAX 232 dengan perancangan skema seperti pada gambar di bawah ini



Gambar 3.7. Rangkaian komunikasi serial RS232

Standar sinyal serial MAX 232 memiliki ketentuan level tegangan sebagai berikut:

1. Logika '1' terletak antara -3 Volt sampai -25 Volt.
2. Logika '0' terletak antara +3 Volt sampai +25 Volt.

Daerah tegangan antara -3 Volt sampai +3 Volt adalah invalid level, level daerah tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga, level tegangan lebih negatif dari -25 Volt atau lebih positif dari +25 Volt juga harus dihindari karena tegangan tersebut dapat merusak line driver pada saluran MAX 232.

### 3.3. Perancangan Perangkat Lunak

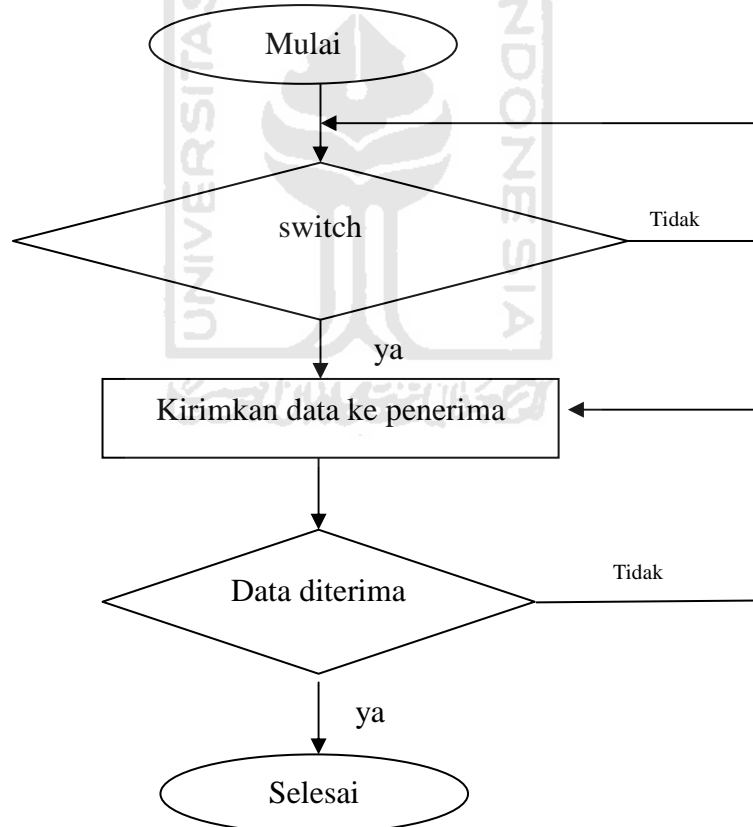
#### 3.3.1. Perancangan Piranti Lunak Mikrokontroler

Untuk menjalankan fungsinya mikrokontroler membutuhkan suatu piranti lunak dengan kode yang akan tersimpan di memori *flash* dari

ATMEGA8535. Piranti aman ini karena tidak menghabiskan terlalu besar kapasitas dari flash mikrokontroler dibandingkan dengan bahasa pemrograman lain. Selain itu bahasa pemrogramannya juga masih mudah untuk dipelajari dan dapat dibaca langsung akan dibuat dengan menggunakan *development Code Vision AVR* yang menggunakan pemrograman bahasa C.

### 3.3.1.1. Piranti Lunak Pengirim data

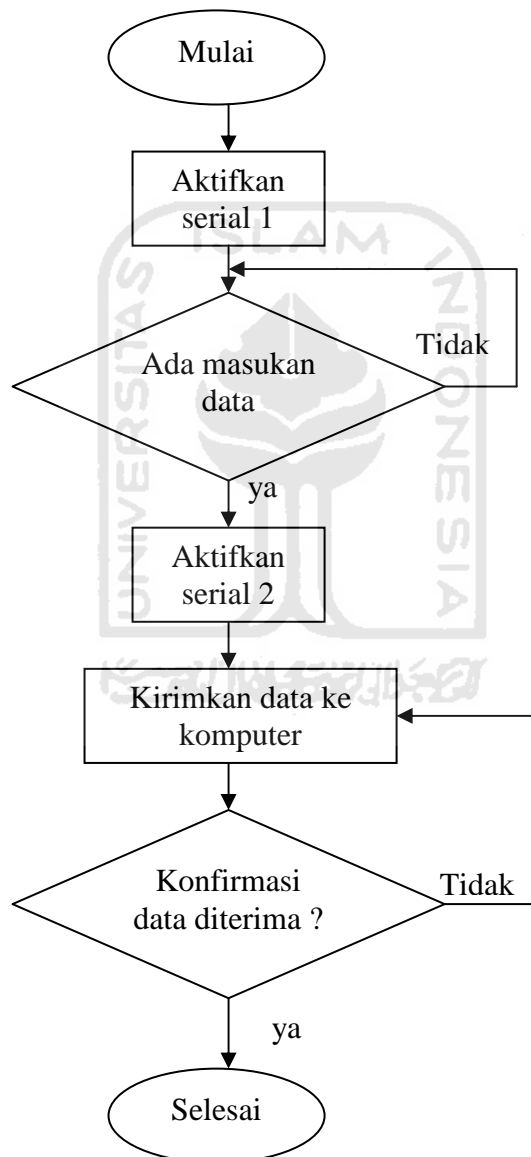
Piranti lunak pengirim data berperan untuk mengolah data untuk dikirimkan ke penerima. Berikut ini adalah diagram alir perancangan piranti lunak pengiriman data.



Gambar 3.8. Diagram alir Piranti Lunak Pengirim Data Mikrokontroler

### 3.3.1.2. Piranti Lunak Antarmuka Komputer

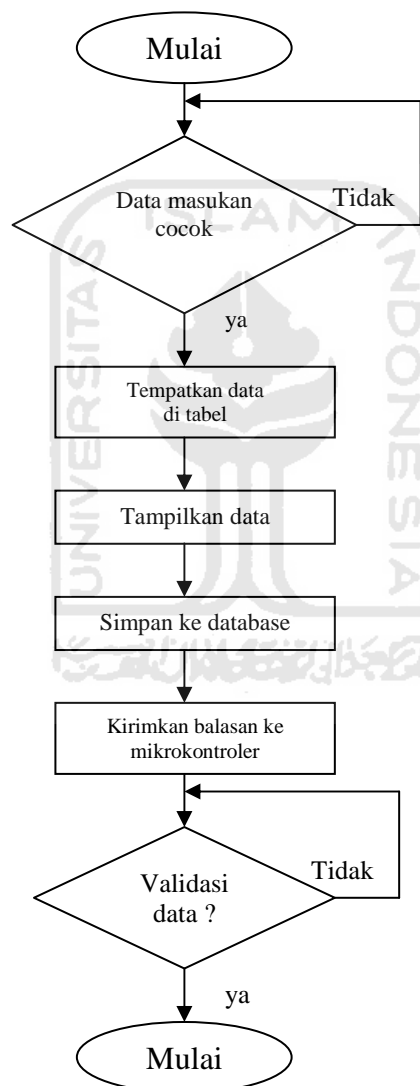
Supaya data dapat diolah oleh komputer maka data dari alat perlu diolah dan dikonversi ke data serial dengan kecepatan data yang lebih tinggi. Oleh karena itu diperlukan suatu piranti lunak yang akan melakukan kontrol. Berikut ini adalah diagram alir dari piranti lunak yang akan dibangun.



Gambar 3.9. Diagram alir Piranti Lunak Antarmuka Mikrokontroler

### 3.3.2. Perancangan Piranti Lunak Komputer

Piranti lunak Komputer dibangun dengan menggunakan bahasa pemrograman *visual basic* dari *Microsoft*. Piranti lunak ini berfungsi untuk melakukan pencatatan data dan menyimpan dalam suatu sistem *database*. Berikut ini adalah diagram alir dari perancangan piranti lunak komputer.



Gambar 3.10. Diagram alir Piranti Lunak Komputer



## BAB IV

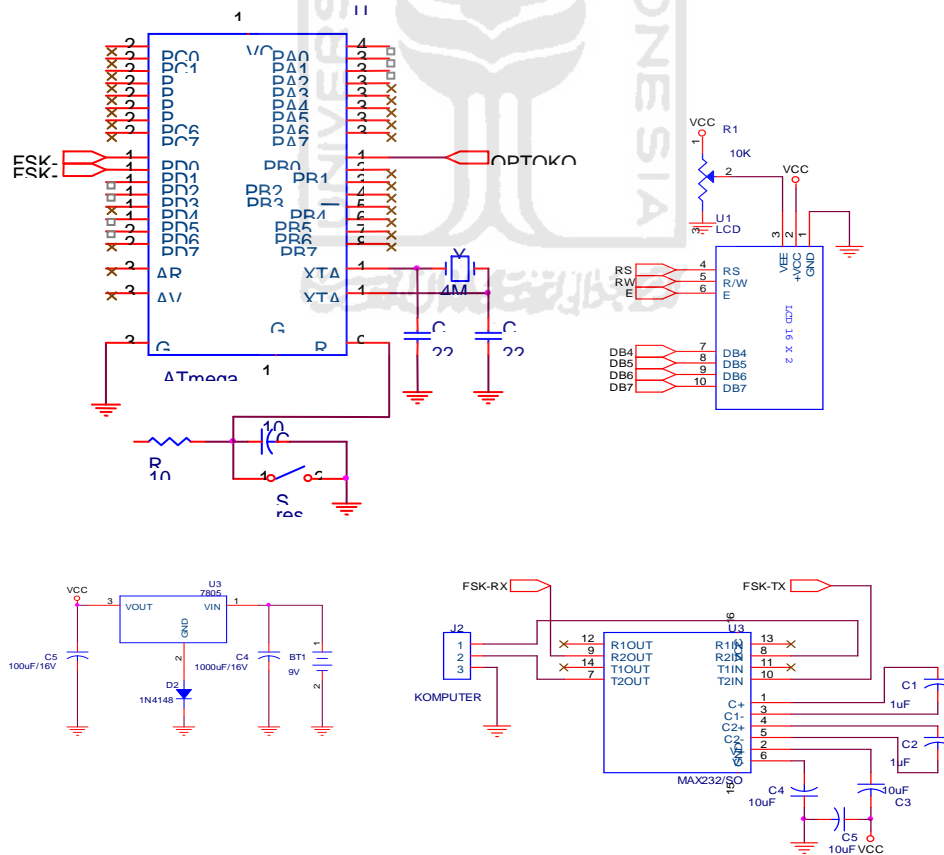
### ANALISIS, DAN PEMBAHASAN

Perancangan alat sistem smart alarm untuk mengatasi pencurian ATM berbasis teknologi seluler dan komputer menghasilkan data dan alat-alat sebagai berikut :

#### 4.1. Hasil Perancangan Piranti Keras

##### 4.1.1. Rangkaian Keseluruhan

Rangkaian hasil dari perancangan pada Bab3 dapat dibuat rangkaian keseluruhan sekaligus sebagai hasil perancangan dari alat yang dapat dibuat guna memenuhi kebutuhan sistem. Berikut rangkaian secara keseluruhan yang telah dibuat ditunjukkan dan pada gambar 4.1. di bawah ini sebagai berikut :



Gambar 4.1. Rangkaian keseluruhan alat

#### 4.1.2. Rangkaian Sensor

Rangkaian sensor berfungsi mengukur tegangan *output* dari optokopler sebagai sensor dengan menghitung jumlah *clock* pulsa yang dihasilkan oleh pergerakan yang berbeda. Tegangan *output* diolah dalam mikrokontroler, dan ditampilkan pada LCD setelah diberi penguatan tegangan dan perubahan menjadi kode biner oleh ADC.

Hasil pengukuran tegangan keluaran sensor optokopler berdasarkan pergerakan yang berbeda diperlihatkan pada Tabel 4.1.

Tabel 4.1. Pengukuran sensor 1 (optokopler)

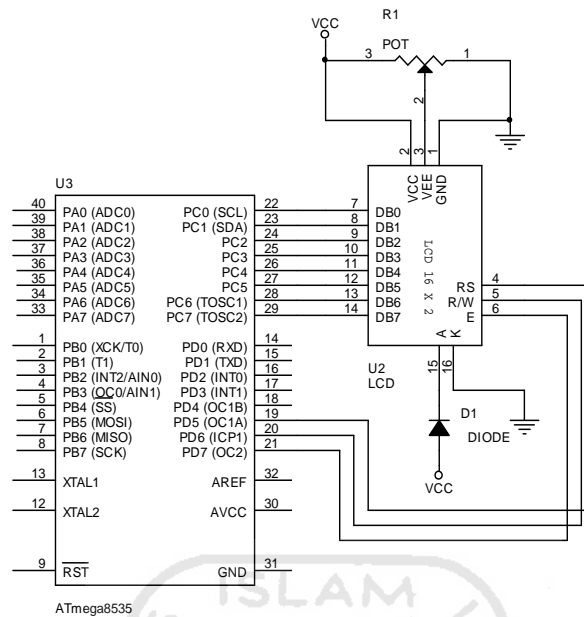
| No | Kondisi        | Hambatan optokopler | Tegangan Output (Volt) |
|----|----------------|---------------------|------------------------|
| 1  | Dihalangi      | 60                  | 0,0                    |
| 2  | Tanpa halangan | 75                  | 4,8                    |

Tabel 4.2. Pengukuran sensor 2 (switch)

| No | Kondisi        | Hambatan switch | Tegangan Output (Volt) |
|----|----------------|-----------------|------------------------|
| 1  | Dihalangi      | 45              | 0,3                    |
| 2  | Tanpa halangan | 60              | 4,7                    |

#### 4.1.3. Rangkaian LCD

Rangkaian penampil LCD ditunjukkan pada Gambar 4.5. di bawah, LCD memiliki tiga bit control RS (*Register Select*), R/W (*Read/Write*) dan E (*Enable*) pin RS dihubungkan ke *port* PD.5, pin R/W dihubungkan ke *port* PD.6 dan pin E dihubungkan ke *port* PD.7 sedangkan 8 bit datanya ( $D_0 - D_7$ ) dihubungkan ke *port* C (PC.0 – PC.7)



Gambar 4.2. Gambar Rangkaian LCD

#### 4.1.4. Catu Daya

Catu daya merupakan piranti keras yang sangat penting dari seluruh sistem karena berfungsi untuk memberi catuan dari seluruh sistem. Gambar berikut ini merupakan hasil dari perancangan catu daya.



Gambar 4.3. Catu Daya

Untuk melihat kinerja dari catu daya perlu dilakukan sedikit pengujian terhadap rangkaian catu daya yang ditunjukkan pada tabel berikut ini :

Tabel 4.3. Pengujian Catu Daya

| No | Waktu     | Tegangan Output catu daya (DC-Volt) | Tegangan listrik sumber (AC-Volt) |
|----|-----------|-------------------------------------|-----------------------------------|
| 1  | 09.00 WIB | 5,2                                 | 205                               |
| 2  | 11.00 WIB | 5,2                                 | 195                               |
| 3  | 13.00 WIB | 5,2                                 | 195                               |
| 4  | 15.00 WIB | 5,2                                 | 190                               |
| 5  | 17.00 WIB | 5,2                                 | 185                               |
| 6  | 19.00 WIB | 5,2                                 | 170                               |

Berdasarkan data di atas tampak bahwa tegangan catu daya tetap stabil walaupun tegangan dari PLN tidak stabil dan cenderung menurun menjelang malam. Pengujian ini ditujukan untuk melihat tingkat kestabilan dari catu daya.

#### 4.2. Hasil Perancangan Piranti Lunak

Setelah dilakukan perancangan, maka telah dihasilkan suatu piranti lunak untuk menampilkan hasil pengukuran dengan menggunakan sensor optokopler. Data yang ditampilkan adalah data antara mikrokontroler dengan sensor.

Piranti lunak dari alat penyortiran barang yang dibangun dibuat dengan mempergunakan *development tool* CodeVision AVR dengan bahasa C Seperti telah dijelaskan pada bab 3 maka listing program secara lebih lengkap dapat dilihat pada lampiran. Berikut ini adalah cuplikan dari listing program yang dibuat.

##### 4.2.1. Inisialisasi Awal

Pada awal program ini yang dilakukan adalah inisialisasi dari frekuensi kristal serta fasilitas-fasilitas yang akan digunakan dalam program, berikut merupakan listing program.

```
#include <mega16.h>
#include <string.h>
#include <delay.h>
```

```
#include <ctype.h>
#asm
    .equ __lcd_port=0x1B
#endasm
#include <lcd.h>
```

#### 4.2.2. Inisialisasi LCD

Selanjutnya dilakukan inisialisasi dari LCD yang akan digunakan dalam perancangan alat dan koneksi ke port I/O mikrokontroler. Berikut ini proses inisialisasi LCD dalam bahasa C, berikut merupakan listing program.

```
void user_data() // USER DATA
{
    z = text_int(); j=8; k=0; m=0; msg_n=0;
    while ( z!=0 )
    {
        if ( j==8 ){ j=0; text_int(); };
        l = ( i>>j ) & 0x01; ++j;
        if ( l==1 ) l=0x80;
        if ( k==7 ){ msg[msg_n++] = m>>1; m=0; k=0; z--; };
        m = m>>1 | l; k++;
    }; msg[msg_n]=0;
}

void display_xxx()
{
    lcd_gotoxy(0,1); i=0;
    while ( xxx[i]!='#' && i<=16 ) lcd_putchar(xxx[i++]);
    while ( i<16 ) ++i, lcd_putchar(' '); delay_ms(500);
}

void display_msg()
{
    i=0; j=0; k=0;
    if ( msg_n > 16 )
    {
        while(i+15 < msg_n){ j=i,k=0,lcd_gotoxy(0,1);
            while(++k<=16) lcd_putchar(msg[j++]); delay_ms(200); ++i;
        };
    }
    else if ( msg_n==0 ) lcd_gotoxy(0,1),lcd_putsf(" < sms empty
> ");
    else { i=16; lcd_gotoxy(0,1);
        while( k < msg_n ) lcd_putchar(msg[k++]),--i;
        while( i != 0 ) lcd_putchar(' '),--i;
    };
    delay_ms(500);
}
```

### 4.3. Pengujian Keseluruhan

Setelah seluruh piranti keras dan piranti lunak selesai dibuat, maka selanjutnya dilakukan pengujian terhadap fungsi keseluruhan rangkaian yang disajikan pada tabel pengujian berikut ini. Pengujian dilakukan dengan menjalankan alat. Pengujian ini ditujukan. untuk mengetahui fungsi atau tidaknya sensor yang terpasang pada alat dan ditunjukkan pada Tabel 4.5.

Tabel 4.4. Pengujian Fungsional Alat

| No | Kegiatan          | Kesimpulan |
|----|-------------------|------------|
| 1  | Sensor optokopler | OK         |
| 2  | Sensor switch     | OK         |
| 3  | LCD               | OK         |
| 4  | Handphone         | OK         |

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Setelah melakukan perancangan terhadap piranti keras dan piranti lunak alat smart alarm untuk mengatasi gangguan keamanan di suatu wilayah berbasis teknologi seluler dan komputer maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Sistem yang dibangun telah dapat berjalan apabila pengirim menghasilkan sinyal keluaran dan handphone menerima sinyal tanpa ada gangguan.
2. Sistem smart alarm dapat dipantau melalui seluler dengan demikian alat telah menunjukkan hasil dengan baik dan berjalan sesuai dengan rancangan.
3. Sistem yang dibangun melalui komunikasi serial berupa seluler dan komputer

#### **5.2. Saran**

Untuk pengembangan perangkat pada penelitian mendatang, penulis mengajukan beberapa masukan sebagai berikut :

1. Dapat dikombinasikan dengan handphone dan GPS menggunakan sensor yang lain selain switch sebagai sensor memiliki kinerja yang lebih baik dibandingkan switch sebagai sensor yang telah dipergunakan pada alat hasil rancangan.

2. Sistem dapat dilengkapi dengan pengendalian melalui lebih dari satu komputer selain itu data hasil pemantauan dapat dicatat dalam database pada komputer sehingga didapatkan hasil yang lebih baik.



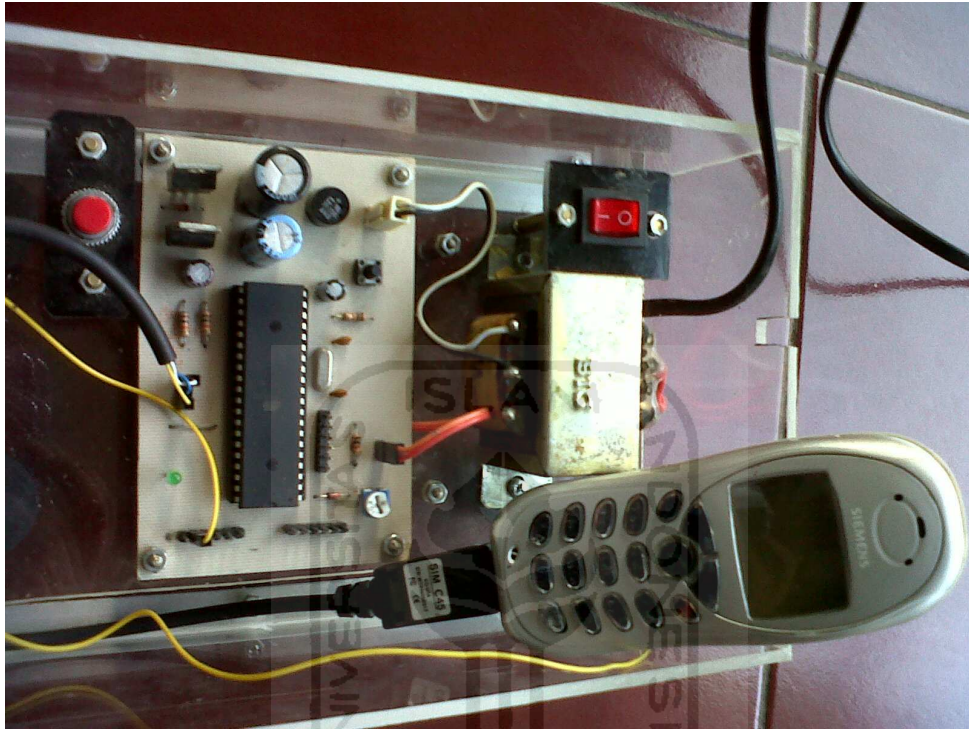


## DAFTAR PUSTAKA

- Anonim, 2006, *ATMega 8535L Datasheet: 8-bit AVR Microcontroller with 8K Bytes m-system Programmable Flash*. Atmel Corporation
- Anthony, Pranata, 1997, *Tip dan Trik Pemrograman Visual Basic*, Andi Offset, Yogyakarta.
- Budiharto, W., 2005, *Perancangan Sistem dan Aplikasi Mikrokontroler*, Penerbit Elex Media Komputindo, Jakarta.
- Budiharto, W., dan Rizal, G., 2007, *Belajar Sendiri 12 Proyek Mikrokontroler Untuk Pemula*, Cetakan kedua, Penerbit Elex Media Komputindo, Jakarta
- Chattopadhyay, D. Dkk., 1989, *Dasar Elektronika*, Penerbit Universitas Indonesia, Jakarta.
- Edminister, Joseph., 1996, *Teori dan Soal – Soal Rangkaian Listrik*, Erlangga
- Malvino, Albert Paul, 2003, *Prinsip-prinsip Elektronika*, Terjemahan Alb Joko Sutoso, Jakarta : Salemba Teknik.
- Mac Kenzie, I.S., 1992, *The 8051 Microcontroller* Second edition, Prentice Hall Inc, New Jersey.
- Prasetia, Retna & Edi Widodo, Catur, 2004, *Teori dan Praktek Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0*, Yogyakarta: Andi Offset.
- Safri, 2002, *Smart Alarm untuk Mobil*, Afiscom.
- Wardana, Lingga, 2006, *Belajar Mikrokontroler AVR seri ATMega8535*, Andi, Yogyakarta
- Wicaksono, M. Hilmi, *Workshop: Introduction to ATMEL Microcontroller*, BiTS (Bijak Training Services).
- William David Cooper, 1985, *Instrumentasi Elektronika dan Teknik Pengukuran*, Erlangga, Jakarta
- Woollard, Barry G, 2003, *Elektronika Praktis*, Terjemahan H. Kristono, Jakarta : Pradnya Paramita.

# LAMPIRAN

## A. FOTO ALAT



## B. LISTING PROGRAM

Listing program mikro

```
/******
```

```
This program was produced by the  
CodeWizardAVR V1.24.0 Standard
```

```
Project : SMS ALARM WIRELESS
```

Date : 28/11/2009

Author : Saiful

Chip type : ATmega32, frequency : 4,0 MHz

\*\*\*\*\*/

```
#include <mega16.h>
```

```
#include <string.h>
```

```
#include <delay.h>
```

```
#include <ctype.h>
```

```
#asm
```

```
    .equ __lcd_port=0x1B
```

```
#endasm
```

```
#include <lcd.h>
```

```
#define buzzer PORTD.7
```

```
bit send,alarm=0,flip,flop,konek=0;
```

```
unsigned char time,tick,tack,tock, buffer[33];
```

```
#define RXB8 1
```

```
#define TXB8 0
```

```
#define UPE 2
```

```
#define OVR 3
```

```
#define FE 4
```

```
#define UDRE 5
```

```
#define RXC 7
```

```
#define FRAMING_ERROR (1<<FE)
```

```
#define PARITY_ERROR (1<<UPE)
```

```
#define DATA_OVERRUN (1<<OVR)
```

```
#define DATA_REGISTER_EMPTY (1<<UDRE)
```

```
#define RX_COMPLETE (1<<RXC)
```

```
#define RX_BUFFER_SIZE 128
```

```
char rx_buffer[RX_BUFFER_SIZE];
```

```
unsigned char data,rx_wr_index,rx_rd_index,rx_counter;
```

```
#pragma savereg-
```

```
interrupt [USART_RXC] void usart_rx_isr(void)
```

```
{
```

```
    char status,data;
```

```
    #asm
```

```

        push r26
        push r27
        push r30
        push r31
        in r26,sreg
        push r26
    #endasm
    status=UCSRA; data=UDR;
    if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
        { rx_buffer[rx_wr_index]=data;
        if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
        if (++rx_counter == RX_BUFFER_SIZE) rx_counter=0;
        konek=1;
        };
    #asm
        pop r26
        out sreg,r26
        pop r31
        pop r30
        pop r27
        pop r26
    #endasm
}
#pragma savereg+

#ifdef _DEBUG_TERMINAL_IO_
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index];
    if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}
#pragma used-
#endif
#include <stdio.h>

```



```

interrupt [EXT_INT2] void ext_int2_isr(void)
{
    alarm=1; buzzer=0; tack=0;
}

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    TCNT1=0x0BDB;
    PORTB.2=1, DDRB.2=0;
    if ( ++tick>=60 ) tick=0, ++tack;
    if ( alarm || PINB.2==0 ) alarm=1, buzzer=0;
    else buzzer=1;
}

void reset_buffer()
{
    #asm("cli")
    konek=0, data=0, rx_counter=0, rx_rd_index=0, rx_wr_index=0;
    #asm("sei")
}

//***** GSM MODEM SUB ROUTINE
*****

flash char key1[9]="REG1 OFF";
flash char key2[8]="REG1 ON";
flash char key3[6]="REG1 ";
flash char key4[9]="REG2 OFF";
flash char key5[8]="REG2 ON";
flash char key6[6]="REG2 ";
flash char key7[9]="REG3 OFF";
flash char key8[8]="REG3 ON";
flash char key9[6]="REG3 ";
flash char keyA[9]="REG4 OFF";
flash char keyB[8]="REG4 ON";
flash char keyC[6]="REG4 ";
flash char keyD[9]="REG5 OFF";
flash char keyE[8]="REG5 ON";
flash char keyF[6]="REG5 ";

flash char keyG[9]="REGISTER";
flash char keyH[6]="TEXT ";

```

```
flash char keyK[6]={"TEXT?"};
flash char keyL[7]={"PULSA "};
flash char keyM[7]={"SMS ON"};
flash char keyN[8]={"SMS OFF"};
```

```
eprom unsigned char sms_en=1,r1_en=1,r2_en=1,r3_en=1,r4_en=1,r5_en=1;
eprom unsigned char reg1[17]={"081328526792#"};
eprom unsigned char reg2[17]={"081328526792#"};
eprom unsigned char reg3[17]={"081328526792#"};
eprom unsigned char reg4[17]={"081328526792#"};
eprom unsigned char reg5[17]={"081328526792#"};
eprom unsigned char save[17]={"081328526792#"};
eprom unsigned char text1[160]={"TOLONG.. !" };
```

```
unsigned char msg_n,pas,i,j,k,l,m,x,y,z;
unsigned char xxx[17],yyy[17]={"081328526792#"};
unsigned char pdu,msg[161];
```

```
void clear_msg()
{ unsigned char i=0; while (i<=160) msg[i++]=0; }
```

```
void kirim_register()
{
```

```
clear_msg(); sprintf(msg,"SETTING: REG1:O "); i=15;
```

```
if ( r1_en==1 ) msg[i++]='N',msg[i++]=' ';
else msg[i++]='F',msg[i++]='F'; msg[i++]=' ';
msg[i++]='T'; msg[i++]='O'; msg[i++]=' ';
j=0; while ( reg1[j]!='#' && i<=16 ) msg[i++]=reg1[j++];
msg[i++]=';'; msg[i++]=' ';
```

```
msg[i++]='R'; msg[i++]='E'; msg[i++]='G'; msg[i++]='2'; msg[i++]=':.';
msg[i++]='O';
```

```
if ( r2_en==1 ) msg[i++]='N',msg[i++]=' ';
else msg[i++]='F',msg[i++]='F'; msg[i++]=' ';
msg[i++]='T'; msg[i++]='O'; msg[i++]=' ';
j=0; while ( reg2[j]!='#' && i<=16 ) msg[i++]=reg2[j++];
msg[i++]=';'; msg[i++]=' ';
```

```

        msg[i++]='R'; msg[i++]='E'; msg[i++]='G'; msg[i++]='2'; msg[i++]='.';
msg[i++]='O';
        if ( r3_en==1 ) msg[i++]='N',msg[i++]=' ' ;
        else msg[i++]='F',msg[i++]='F'; msg[i++]=' ' ;
        msg[i++]='T'; msg[i++]='O'; msg[i++]=' ' ;
        j=0; while ( reg3[j]!='#' && i<=16 ) msg[i++]=reg3[j++];
        msg[i++]='.'; msg[i++]=' ' ;

        msg[i++]='R'; msg[i++]='E'; msg[i++]='G'; msg[i++]='2'; msg[i++]='.';
msg[i++]='O';
        if ( r4_en==1 ) msg[i++]='N',msg[i++]=' ' ;
        else msg[i++]='F',msg[i++]='F'; msg[i++]=' ' ;
        msg[i++]='T'; msg[i++]='O'; msg[i++]=' ' ;
        j=0; while ( reg4[j]!='#' && i<=16 ) msg[i++]=reg4[j++];
        msg[i++]='.'; msg[i++]=' ' ;

        msg[i++]='R'; msg[i++]='E'; msg[i++]='G'; msg[i++]='2'; msg[i++]='.';
msg[i++]='O';
        if ( r5_en==1 ) msg[i++]='N',msg[i++]=' ' ;
        else msg[i++]='F',msg[i++]='F'; msg[i++]=' ' ;
        msg[i++]='T'; msg[i++]='O'; msg[i++]=' ' ;
        j=0; while ( reg5[j]!='#' && i<=16 ) msg[i++]=reg5[j++];
        msg[i++]='.'; msg[i++]=' ' ;

        msg[i++]='.'; msg_n=i;
}

void load_text1()
{
    clear_msg(); i=0; j=0;
    while ( text1[j]!=0xFF && text1[j]!=0 && i<=150 )
        msg[i++]=text1[j++]; msg_n=i;
}

void data_text()
{
    clear_msg();
    sprintf(msg,"TEXT:  "); i=6;
    j=0; while ( text1[j]!=0xFF && text1[j]!=0 && i<=159 )
        msg[i++]=text1[j++]; msg_n=i;
}

```

```

unsigned char modem_chek()
{
    unsigned char i;
    konek=0; putchar(0x1B);
    printf("AT\r\n"); delay_ms(200);
    if ( !konek ) lcd_clear(),
    lcd_putsf(" MODEM CHEK\n < no modem > "),
    i=0, delay_ms(1000);
    else i=1; return i;
}

unsigned char response()
{
    unsigned char i=0, j=0;
    do { getchar();
        if ( data=='R' ) j=1, i=0;
        if ( data=='K' ) j=1, i=1;
    } while( j==0 ); return i;
}

void delete_sms( unsigned char del )
{
    delay_ms(200);
    do { printf("AT+CMGD=%d\r\n", del), delay_ms(250);
        } while( response() == 0 );
    lcd_gotoxy(0,1), lcd_putsf(" < delete sms > "), delay_ms(250);
}

void kirim( unsigned char x ) // SERIAL DATA TRANSMIT
{
    pdu++; if ( pas==0 ) printf( "%02X", x ); }

void pdu_encoder() // ENCODER SMS PDU MODE > TEXT
STRING
{
    unsigned char i,j,k,l,m;
    kirim( 0x00 ); pdu=0; // SC > SERVICE CENTRE
    kirim( 0x11 ); kirim( 0x00 ); i=0; j=1; // FO > FIRST OCTET, MR >
MEMORY REFERENCE

    if ( yyy[0]=='0' ) j=0; // LENGHT OF DESTINATION
ADDRESS ( LODA )

```



```

while ( yyy[i]!='#' ) ++i,++j;
if ( yyy[0]=='+' ) j-=2,m=0x91; // TODA ( TYPE OF DA )
else m = 0x81; kirim( j ); kirim( m ); // 81 = NASIONAL, 91 =
INTERNASIONAL

i = j/2; ++i; k=0; // DA ADDRESS
if ( yyy[0]=='+' ) k=1;
do { if ( pas==0 )
{ ++k; putchar( yyy[k--]); //--k;
putchar( yyy[k++]); ++k; //++k;
}; i--; ++pdu;
} while ( i!=0 && i!=1 );

if ( (j%2)!=0 )
{ if ( pas==0 )
{ putchar('F'); putchar( yyy[k]); }; ++pdu;
}; kirim(0),kirim(0),kirim(0xFF),kirim(msg_n); // PID DCS VP UDL

i=0xFF,j=7,k=0,l=0,m=0; // USER DATA ( THE
MESSAGE )
do { if (j==7) j=0, ++i;
l=(msg[i]>>j)&0x01, ++j;
if (l==1) l=0x80; else l=0;
if (k==8) kirim(m), m=0, k=0;
m=(m>>1)|l,++k;
} while (i<=msg_n||k!=8);
}

void display_yyy()
{
unsigned char i=0;
lcd_gotoxy(0,1);
while ( yyy[i]!='#' && i<=16 ) lcd_putchar(yyy[i++]);
while ( i<16 ) ++i, lcd_putchar(' ');
}

void kirim_sms( unsigned char T ) // SMS SENDER N RETRY
{
if ( modem_chek()==1 )
{
if ( sms_en==1 )
{

```

```

while ( T!=0 )
{
    lcd_clear(); lcd_putsf("SMS SEND TO:");
    display_yyy(); delay_ms(500);
    putchar(0x1B); // ESCAPE ( 0x1B )
    pas=1, pdu_encoder(); // PDU COUNTER
    printf("AT+CMGS=%d\r\n",pdu);
    while ( getchar()!='>' );
    pas=0, pdu_encoder(), putchar(0x1A); // CTR+Z

    if ( response()==1 )
    { T=0,lcd_gotoxy(0,1),lcd_putsf("< success >"); }
    else
    { if (T==0) sprintf(buffer,"< send fail >");
      else sprintf( buffer,"< retry = %d >",T);
      lcd_gotoxy(0,1); lcd_puts(buffer); delay_ms(1500); --T;
    };
};
} else lcd_clear(), lcd_putsf(" SMS SENDER\n IS DISSABLED ! ");
delay_ms(1000);
}; reset_buffer();
}

```

```

void skipchar2(){ getchar(); getchar(); }
void skipchar4(){ skipchar2(); skipchar2(); }
void load_reg1(){ i=0; while ( i<=16 ) yyy[i]=reg1[i++]; yyy[i]='#'; }
void load_reg2(){ i=0; while ( i<=16 ) yyy[i]=reg2[i++]; yyy[i]='#'; }
void load_reg3(){ i=0; while ( i<=16 ) yyy[i]=reg3[i++]; yyy[i]='#'; }
void load_reg4(){ i=0; while ( i<=16 ) yyy[i]=reg4[i++]; yyy[i]='#'; }
void load_reg5(){ i=0; while ( i<=16 ) yyy[i]=reg5[i++]; yyy[i]='#'; }

```

```

void load_save(){ i=0; while ( i<=16 ) yyy[i]=save[i++]; yyy[i]='#'; }

```

```

unsigned char text_int()
{ i=(unsigned char)(toint(getchar())<<4) + toint(getchar()); return i; }

```

```

unsigned char desimal()
{ i=0; while ( isalnum(getchar())) i=(i*10)+toint(data); return i; }

```

```

void time_stamp()

```

```

{      i=0; while( ++i<=7 ) skipchar2(); }

void sender()                                // ORIGINATE ADDRESS
{      l = text_int(); text_int(); k=0;
      if ( i==0x91 ) save[k]='+',xxx[k++]='+';
      while( l!=0 && l!=1 )
      { xxx[++k]=getchar(); save[k]=data; xxx[--k]=getchar(); save[k]=data;
k+=2,l-=2; };
      if ( l==1 ) getchar(),xxx[k]=getchar(),save[k++]='data';
      xxx[k]='#'; save[k]='#';
}

void user_data()                             // USER DATA
{      z = text_int(); j=8; k=0; m=0; msg_n=0;
      while ( z!=0 )
      { if ( j==8 ){ j=0; text_int(); };
        l=( i>>j)& 0x01; ++j;
        if ( l==1 ) l=0x80;
        if ( k==7 ){ msg[msg_n++]='m>>1'; m=0; k=0; z--; };
        m = m>>1|l; k++;
      }; msg[msg_n]=0;
}

void display_xxx()
{      lcd_gotoxy(0,1); i=0;
      while ( xxx[i]!='#' && i<=16 ) lcd_putchar(xxx[i++]);
      while ( i<16 ) ++i, lcd_putchar(' '); delay_ms(500);
}

void display_msg()
{
      i=0; j=0; k=0;
      if ( msg_n > 16 )
      { while(i+15 < msg_n){ j=i,k=0,lcd_gotoxy(0,1);
        while(++k<=16) lcd_putchar(msg[j++]); delay_ms(200); ++i; };
      }
      else if ( msg_n==0 ) lcd_gotoxy(0,1),lcd_putsf(" < sms empty > ");
      else { i=16; lcd_gotoxy(0,1);
      while( k < msg_n ) lcd_putchar(msg[k++]),--i;
      while( i != 0 ) lcd_putchar(' '),--i;
      };
      delay_ms(500);
}

```

```

}
void pdu_decoder()
{
    msg_n=0; clear_msg(); // clear message buffer
    text_int();
    while( i!=0){ skipchar2(); --i; } // Service centre
    i = text_int() & 0x03; // FO => MT1 MT2
    if ( i==0 ) // 0x00 > SMS DELIVER
    { sender(); skipchar4(); time_stamp();
      user_data(); i=1; }
    else if ( i==2 ) // 0x10 > SMS STATUS REPORT
    { skipchar2(); sender(); time_stamp();
      time_stamp(); i=0; }
    else if ( i==1 ) // 0x01 > SMS SUBMIT STORED
    { skipchar2(); sender(); skipchar4();
      skipchar2(); user_data(); i=1; };

    i=0; while ( i<=160 ) msg[i]=toupper(msg[i++]);
    if ( i ) display_xxx(), display_msg();
}

unsigned char cek_pulsa()
{
    k=5; while ( k > 0 )
    { delay_ms(500);
      clear_msg(); reset_buffer();
    i=0; while (buffer[i]!='@')
      putchar(buffer[i++]); printf(";\r\n");
    while (getchar()!=10); i=0; l=0;
    while ( i==0 && l==0 )
    { getchar();
      if ( data=='E' ) i=1;
      else if ( data=='N' ) i=1;
      else if ( data=="" || data=='S' )
      { i=0; msg[i++]=toupper( data );
        while ( getchar()!=13 && i<=120 && data!='' && data!=';')
          msg[i++]=toupper( data ); msg[i++]=' ',
          msg[i++]='.',msg[i++]='.',msg[i++]='.',msg_n=i;
        i=0; while ( ++i<=100 )
        { if ( msg[i]=='P'
          && msg[(unsigned char)(i+1)]=='U'

```

```

        && msg[(unsigned char)(i+2)]=='L'
        && msg[(unsigned char)(i+3)]=='S'
        && msg[(unsigned char)(i+4)]=='A' ) k=0, l=1;
    }; i=1;
    } else --k,l=0,i=1;
    delay_ms(3000); /*reset_buffer()*/ i=1;
};
}; return l;
}

```

```
void cek_sms()
```

```
{ unsigned char index=1,send_sistem=0,send_txt=0;
```

```
if ( modem_chek()==1 )
```

```
{
```

```
    i=0,x=0,y=0;
```

```
    printf("AT+CPMS?\r\n");
```

```
    while ( i==0 )
```

```
    { if ( getchar()=='.' )
```

```
        { while ( getchar()!='.' );
```

```
          x=desimal(),y=desimal(),i=1;
```

```
          while ( getchar()!=13 ); }
```

```
    else if ( data=='R' ) i=1;
```

```
};
```

```
while ( x!=0 && index!=(y+1) )
```

```
{
```

```
    send_sistem=0,send_txt=0; clear_msg();
```

```
    z=0; sprintf(buffer,"SMS INPUT NO. %d",index );
```

```
    lcd_clear(), lcd_puts(buffer);
```

```
    printf("AT+CMGR=%d\r\n", index );
```

```
    while( getchar()!='.' ); while( getchar()!='.' );
```

```
    while( isalnum(getchar())) z=(z*10)+toint(data);
```

```
    while( getchar()!=10 );
```

```
    if ( z!=0 )
```

```
    { pdu_decoder(),--x;
```

```
      delete_sms(index);
```

```
    lcd_gotoxy(0,1);
```

```
    if ( strncmpf( msg,key1,8 )==0 )
```

```
        // MAMATIKAN REGISTER 1
```

```

    { lcd_putsf("< REGISTER1 OFF>");
      r1_en=0; send_sistem=1;
    }
else if ( strncmpf( msg,key2,7 )==0 ) // MENGAKTIFKAN REGISTER 1
    { lcd_putsf("< REGISTER1 ON >");
      r1_en=1; send_sistem=1;
    }
else if ( strncmpf( msg,key3,5 )==0 ) // MERUBAH REGISTER 1
    { i=0; j=5; reg1[i++]=msg[j++];
      while ( isalnum(msg[j]) )
        reg1[i++]=msg[j++]; reg1[i++]='#';
      lcd_clear(); lcd_putsf("SET REGISTER1: \n");
      i=0; while ( i<16 && reg1[i]!='#')
        lcd_putchar( reg1[i++]); r1_en=1;
      send_sistem=1;
    }

else if ( strncmpf( msg,key4,8 )==0 ) // MAMATIKAN REGISTER 2
    { lcd_putsf("< REGISTER2 OFF>");
      r2_en=0; send_sistem=1;
    }
else if ( strncmpf( msg,key5,7 )==0 ) // MENGAKTIFKAN REGISTER 2
    { lcd_putsf("< REGISTER2 ON >");
      r2_en=1; send_sistem=1;
    }
else if ( strncmpf( msg,key6,5 )==0 ) // MERUBAH REGISTER 2
    { i=0; j=5; reg2[i++]=msg[j++];
      while ( isalnum(msg[j]) )
        reg2[i++]=msg[j++]; reg2[i++]='#';
      lcd_clear(); lcd_putsf("SET REGISTER2: \n");
      i=0; while ( i<16 && reg2[i]!='#')
        lcd_putchar( reg2[i++]); r2_en=1;
      send_sistem=1;
    }

else if ( strncmpf( msg,key7,8 )==0 ) // MAMATIKAN REGISTER 3
    { lcd_putsf("< REGISTER3 OFF>");
      r3_en=0; send_sistem=1;
    }
else if ( strncmpf( msg,key8,7 )==0 ) // MENGAKTIFKAN REGISTER 3
    { lcd_putsf("< REGISTER3 ON >");
      r3_en=1; send_sistem=1;
    }

```

```

    }
else if ( strncmppf( msg,key9,5 )==0 ) // MERUBAH REGISTER 3
    { i=0; j=5; reg3[i++]=msg[j++];
      while ( isalnum(msg[j]) )
        reg3[i++]=msg[j++]; reg3[i++]='#';
      lcd_clear(); lcd_putsf("SET REGISTER3: \n");
      i=0; while ( i<16 && reg3[i]!='#')
        lcd_putchar( reg3[i++]); r3_en=1;
      send_sistem=1;
    }

else if ( strncmppf( msg,keyA,8 )==0 ) // MAMATIKAN REGISTER 4
    { lcd_putsf("< REGISTER4 OFF>");
      r4_en=0; send_sistem=1;
    }
else if ( strncmppf( msg,keyB,7 )==0 )// MENGAKTIFKAN REGISTER 4
    { lcd_putsf("< REGISTER4 ON >");
      r4_en=1; send_sistem=1;
    }
else if ( strncmppf( msg,keyC,5 )==0 )// MERUBAH REGISTER 4
    { i=0; j=5; reg4[i++]=msg[j++];
      while ( isalnum(msg[j]) )
        reg4[i++]=msg[j++]; reg4[i++]='#';
      lcd_clear(); lcd_putsf("SET REGISTER4: \n");
      i=0; while ( i<16 && reg4[i]!='#')
        lcd_putchar( reg4[i++]); r4_en=1;
      send_sistem=1;
    }

else if ( strncmppf( msg,keyD,8 )==0 ) // MAMATIKAN REGISTER 5
    { lcd_putsf("< REGISTER5 OFF>");
      r5_en=0; send_sistem=1;
    }
else if ( strncmppf( msg,keyE,7 )==0 )// MENGAKTIFKAN REGISTER 5
    { lcd_putsf("< REGISTER5 ON >");
      r5_en=1; send_sistem=1;
    }
else if ( strncmppf( msg,keyF,5 )==0 )// MERUBAH REGISTER 5
    { i=0; j=5; reg5[i++]=msg[j++];
      while ( isalnum(msg[j]) )
        reg5[i++]=msg[j++]; reg5[i++]='#';
      lcd_clear(); lcd_putsf("SET REGISTER5: \n");

```

```

        i=0; while ( i<16 && reg5[i]!='#)
        lcd_putchar( reg5[i++]); r5_en=1;
        send_sistem=1;
    }

    else if ( strncmppf( msg,keyG,8 )==0 )                // MINTA DATA
REGISTER
    { lcd_clear(),lcd_putsf("< REGISTER ? >");
    send_sistem=1;
    }

    else if ( strncmppf( msg,keyH,5 )==0 )                // GANTI TEXT 1
    { lcd_clear(),lcd_putsf("< SET SMS TEXT >");
    i=0; j=5; while ( msg[j]!=0 && i<=150 )
    text1[i++]=msg[j++]; text1[i++]=0xFF;
    send_txt=1;
    }

    else if ( strncmppf( msg,keyK,5 )==0 )                // MINTA DATA TEXT
    { lcd_clear(),lcd_putsf("< REQUEST TEXT >");
    send_txt=1;
    }

    else if ( strncmppf( msg,keyL,6 )==0 )                // MINTA DATA SISAPULSA
    { if ( modem_chek()==1 )
    { load_save(); lcd_clear();
    lcd_putsf(" SALDO PULSA\n DIAL TO ");
    lcd_gotoxy(9,1);
    i=0; while ( i<=32) buffer[i++]=0; // EX. CODE = <*123#>
    buffer[0]='A',buffer[1]='T',buffer[2]='D';// KE PENGIRIM SMS
    i=3; j=6; while ( msg[j]!=0 && i<=10 )
    lcd_putchar(msg[j]), buffer[i++]=msg[j++];
    buffer[i++]='@';
    if ( cek_pulsa()==1 ) kirim_sms(5);
    };
    }

    else if ( strncmppf( msg,keyM,6 )==0 )                // GLOBAL SMS
SENDER > ON
    { lcd_putsf("< SET SMS: ON >");
    sms_en=1; send_sistem=1;
    }

```



```

        else if ( strcmpf( msg,keyN,7 )==0 )                // GLOBAL SMS
SENDER > OFF
    { lcd_putsf("< SET SMS: OFF >");
      delay_ms(1000); sms_en=0;
    };

};

++index;
if ( send_txt!=0 )
{ delay_ms(1000); send_txt=0, load_save(), data_text(), kirim_sms(5); };
if ( send_sistem!=0 )
{ delay_ms(1000); send_sistem=0, load_save(), kirim_register(), kirim_sms(5);
};
delay_ms(250);

};
}; reset_buffer();
}

void main(void)
{
    PORTA=0xFF; DDRA=0xFF;
    PORTB=0xFF; DDRB=0x00;
    PORTC=0xFF; DDRC=0xFF;
    PORTD=0x0F; DDRD=0xFE;
    TCCR1A=0x00; TCCR1B=0x03; // Timer/Counter 1 initialization
    TCNT1H=0x0B; TCNT1L=0xDB; // Clock value: 62,500 kHz
    OCR1AH=0x00; OCR1AL=0x00;
    OCR1BH=0x00; OCR1BL=0x00;
    GICR|=0x20; MCUCR=0x00; // INT2 Mode: Falling Edge
    MCUCSR=0x00; GIFR=0x20;
    TIMSK=0x04; // Timer Interrupt initialization

    UCSRA=0x00; UCSRB=0x98; // USART Baud rate: 19200
    UCSRC=0x86; UBRRH=0x00;
    UBRRL=0x0C;
    ACSR=0x80; SFIOR=0x00; // Analog Comparator Output: Off

    lcd_init(16); lcd_clear();

```

```
    lcd_putsf(" SMART ALARM\n WARNING SYSTEM "); delay_ms(1000);  
    lcd_clear(); lcd_putsf("TEGUH SUGIYANTO\n U.T.Y. 2009");  
    delay_ms(500); buzzer=1; delay_ms(100); buzzer=0; delay_ms(500);
```

```
    reg1[16]='#'; reg2[16]='#'; reg3[16]='#'; reg4[16]='#'; reg5[16]='#';  
    xxx[16]='#'; yyy[16]='#'; save[16]='#';  
    text1[159]=0xFF; tick=0; tack=10;  
    send=0; alarm=0; buzzer=1; time=0;
```

```
    lcd_clear();  
    if ( r1_en==1 ) lcd_putsf("REGISTER 1 > ON\n");  
    else lcd_putsf("REGISTER 1 > OFF\n");  
    i=0; while ( i<=16 && reg1[i]!='#' ) lcd_putchar( reg1[i++] ); delay_ms(1000);
```

```
    lcd_clear();  
    if ( r2_en==1 ) lcd_putsf("REGISTER 2 > ON\n");  
    else lcd_putsf("REGISTER 2 > OFF\n");  
    i=0; while ( i<=16 && reg2[i]!='#' ) lcd_putchar( reg2[i++] ); delay_ms(1000);
```

```
    lcd_clear();  
    if ( r3_en==1 ) lcd_putsf("REGISTER 3 > ON\n");  
    else lcd_putsf("REGISTER 3 > OFF\n");  
    i=0; while ( i<=16 && reg3[i]!='#' ) lcd_putchar( reg3[i++] ); delay_ms(1000);
```

```
    lcd_clear();  
    if ( r4_en==1 ) lcd_putsf("REGISTER 4 > ON\n");  
    else lcd_putsf("REGISTER 4 > OFF\n");  
    i=0; while ( i<=16 && reg4[i]!='#' ) lcd_putchar( reg4[i++] ); delay_ms(1000);
```

```
    lcd_clear();  
    if ( r5_en==1 ) lcd_putsf("REGISTER 5 > ON\n");  
    else lcd_putsf("REGISTER 5 > OFF\n");  
    i=0; while ( i<=16 && reg5[i]!='#' ) lcd_putchar( reg5[i++] ); delay_ms(1000);
```

```
    #asm("sei")
```

```
while (1)  
{  
    lcd_clear(); lcd_putsf(" = STAND BY = ");  
    if ( alarm!=1 )  
    { if ( flop ) { if ( --tock>16 ) tock=0,flop=0; }  
      else { if ( ++tock>=15 ) tock=14,flop=1; };
```

```

    lcd_gotoxy(tock,1); lcd_putchar(0xDB); lcd_putchar(0xDB); delay_ms(100);
};

if ( alarm==1 )
{ flip = ~flip; lcd_clear();
  if ( flip ) lcd_putsf(" = ALARM ON = "); lcd_gotoxy(0,1);
  sprintf(buffer," time out %02d", 60-tick );
  lcd_puts(buffer); delay_ms(250);
  if ( !send )
  { if ( r1_en==1 ) load_reg1(),load_text1(), kirim_sms(5),send=1,tick=0;
    if ( r2_en==1 ) load_reg2(),load_text1(), kirim_sms(5),send=1,tick=0;
    if ( r3_en==1 ) load_reg3(),load_text1(), kirim_sms(5),send=1,tick=0;
    if ( r4_en==1 ) load_reg4(),load_text1(), kirim_sms(5),send=1,tick=0;
    if ( r5_en==1 ) load_reg5(),load_text1(), kirim_sms(5),send=1,tick=0;
  };
  } else time=0, send=0, flip=1;

  if ( rx_counter!=0 ){ if ( getchar()=='+' ) tack=10; };
if ( tack !=0 ) alarm=0, send=0, cek_sms(), tack=0;

};
}

```

