

**PERANCANGAN DAN PEMBUATAN SOFTWARE PENGENDALI
PINTU PAGAR OTOMATIS MENGGUNAKAN SMS DAN KODE PIN
BERBASIS MIKROKONTROLER AT89S52**

PROYEK AKHIR

*Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Program Study D III
Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang*



Oleh :

DESSY WULAN SARI

2008/03283

Konsentrasi : Teknologi Sistem Komputer

PROGRAM STUDI TEKNIK ELEKTRONIKA

JURUSAN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2012

HALAMAN PERSETUJUAN PROYEK AKHIR

Judul : Perancangan dan Pembuatan Software Pengendali
Pintu Pagar Otomatis Menggunakan SMS dan Kode
PIN Berbasis Mikrokontroler AT89S52

Nama : Dessy Wulan Sari

BP / NIM : 2008 / 03283

Jurusan : Teknik Elektronika

Program Studi : Diploma III

Fakultas : Teknik

Padang, Januari 2012

Disetujui oleh :
Dosen Pembimbing

Drs. Almasri, MT
NIP. 19640713 198803 1 016

Mengetahui,
Ketua Jurusan Teknik Elektronika
Fakultas Teknik Universitas Negeri Padang

Drs. Putra Jaya, MT
NIP. 19621020 198602 1 001

HALAMAN PENGESAHAN LULUS UJIAN

PROYEK AKHIR

Dinyatakan Lulus Setelah Dipertahankan

Di Depan Tim Penguji Proyek Akhir Jurusan Teknik Elektronika

Fakultas Teknik Universitas Negeri Padang

Judul : Perancangan dan Pembuatan Software Pengendali
Pintu Pagar Otomatis Menggunakan SMS dan Kode
PIN Berbasis Mikrokontroler AT89S52

Nama : Dessy Wulan Sari

BP / NIM : 2008 / 03283

Jurusan : Teknik Elektronika

Program Studi : Diploma III

Fakultas : Teknik

Padang, Januari 2012

Tim Penguji :

	Nama	Pengesahan
Ketua	: Drs. Zulkifli Naansah	1. _____
Sekretaris	: Drs. Almasri, MT	2. _____
Anggota	: Drs. Putra Jaya, MT	3. _____

ABSTRAK

Dessy Wulan Sari (2008/ 03283) : Perancangan dan Pembuatan Software Pengendali Pintu Pagar Otomatis Menggunakan SMS dan Kode PIN Berbasis Mikrokontroler AT89S52

Pintu Pagar Otomatis menggunakan SMS dan Kode PIN berbasis Mikrokontroler AT89S52 ini dibuat untuk kemudahan dalam hidup manusia terutama dalam membuka dan menutup pintu pagar. Pintu pagar yang dikendalikan secara manual dirasakan masih kurang efektif dan efisien sehingga tidak dapat bekerja lebih optimal. Sehingga di buat pintu pagar secara otomatis melalui SMS dan kode PIN berbasis mikrokontroler AT89S52. Tujuan dari pembuatan alat tersebut adalah untuk menghasilkan sebuah *software* alat kontrol secara otomatis yang dapat diimplementasikan untuk membuka dan menutup pagar berbasis mikrokontroler AT89S52 menggunakan SMS dan kode PIN serta proses pengiriman dan penerimaan password dari *keypad* sebagai input data yang ditampilkan pada LCD untuk membuka dan menutup pintu pagar.

Software pengendali Pintu Pagar Otomatis menggunakan SMS dan Kode PIN berbasis Mikrokontroler AT89S52 di buat menggunakan bahasa *assembly*. Sistem ini dikendalikan melalui pesan singkat atau yang dikenal dengan nama SMS (*Short Massage Service*) dan kode PIN sebagai input data yang ditampilkan pada LCD (*Lyquid Cristal Display*).

Dari hasil pengujian yang dilakukan, *user* dapat membuka atau menutup pintu pagar dari jarak jauh dengan mengirimkan SMS sesuai dengan format yang telah ditentukan, SMS tersebut dikirimkan ke ponsel *slave* yang telah terhubung dengan rangkaian mikrokontroler, setelah diterima SMS tersebut diinisialisasi oleh mikrokontroler, jika SMS tersebut mengandung nilai yang sama dengan perintah untuk membuka pintu pagar maka pintu pagar akan terbuka, dan sebaliknya jika SMS tersebut perintah untuk menutup pintu pagar maka pintu pagar akan tertutup.

Kata kunci : Mikrokontroler AT89S52, *Hand phone*, SMS (*Short Massage Service*), LCD (*Lyquid Cristal Display*), Bahasa *Assembly*.

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Allah SWT yang telah meninggikan derajat orang-orang yang beriman dan berilmu pengetahuan, atas berkat rahmat dan karunia-Nya penulis dapat menyelesaikan Proyek Akhir yang berjudul **“Perancangan dan Pembuatan Software Pengendali Pintu Pagar Otomatis Menggunakan SMS dan Kode PIN Berbasis Mikrokontroler AT89S52”**. Selanjutnya shalawat beserta salam semoga disampaikan Allah SWT kepada junjungan kita Nabi Muhammad SAW yang menjadi suri tauladan dalam setiap sikap dan tindakan sebagai seorang muslim.

Pembuatan Proyek Akhir ini merupakan salah satu syarat dalam menyelesaikan pendidikan Diploma Tiga (D3) Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.

Penyelesaian Proyek Akhir ini tidak terlepas dari bantuan dan bimbingan serta dorongan dari berbagai pihak sehingga dapat diselesaikan segala hambatan dan rintangan yang dihadapi, untuk itu penulis mengucapkan terima kasih kepada:

1. Bapak Drs. Ganefri, M.Pd, P.hd, selaku Dekan Fakultas Teknik Universitas Negeri Padang.
2. Bapak Drs. Putra Jaya, MT, selaku Ketua Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang, sekaligus dosen penguji.
3. Bapak Yasdinul Huda, S.Pd, MT, selaku Sekretaris Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.

4. Bapak Drs. Almasri, MT, selaku Ketua Program Studi Jurusan Teknik Elektronika, sekaligus dosen Pembimbing yang telah banyak memberikan masukan dan pemikiran dalam penyelesaian Proyek Akhir ini.
5. Bapak Dedy Irfan, S.Pd, M.Kom, selaku Penasehat Akademis.
6. Bapak Drs. Zulkifli Naansah selaku dosen penguji.
7. Ibu dan Ayah tercinta, adik-adikku yang selalu memberi dorongan serta kasih sayang.
8. Seluruh Staf Pengajar beserta Teknisi Labor Jurusan Teknik Elektronika.
9. Teman - teman seperjuangan Fakultas Teknik Universitas Negeri Padang yang turut membantu dan memberi semangat dalam penyelesaian Proyek Akhir ini.

Penulis menyadari bahwa Proyek Akhir ini masih jauh dari sempurna, karena tidak ada yang sempurna di dunia ini selain Allah SWT. Penulis sangat berharap kepada pembaca untuk memberikan kritik dan saran yang bersifat membangun untuk kemungkinan pengembangan Proyek Akhir ini.

Penulis berharap semoga Proyek Akhir ini bermanfaat bagi semua pihak dan bernilai ibadah di sisi Allah SWT.

Padang, Januari 2012

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSEMBAHAN	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Identifikasi Masalah	2
C. Batasan Masalah	3
D. Perumusan Masalah	3
E. Tujuan Proyek Akhir	4
F. Manfaat Proyek Akhir	4
BAB II LANDASAN TEORI	
A. Perangkat Lunak (<i>Software</i>)	5
B. Mikrokontroler AT89S52	5
C. Interupsi pada Mikrokontroler AT89S52	9
D. <i>Special Function Register</i> (SFR)	9
E. Pemrograman Mikrokontroler AT89S52	12
F. Bahasa <i>Assembler</i>	17
G. Teknologi SMS (<i>Short Message Service</i>)	27
H. <i>AT Command</i>	31
BAB III PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK	
A. Blok Diagram	33

B. <i>Flow Chart</i> Diagram	34
C. Prinsip Kerja	36
D. Rancangan Perangkat Lunak (<i>Software</i>)	37
BAB IV PENGUJIAN DAN ANALISA PROGRAM	
A. Langkah Pengujian Program	41
B. Analisa Program	41
C. Pengoperasian Alat	61
BAB V PENUTUP	
A. Kesimpulan	63
B. Saran	64
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR TABEL

Tabel	Halaman
1. Pin-pin Khusus pada Port 3 Mikrokontroler AT89S52	8
2. Contoh <i>AT Command</i> pada <i>HP SIEMENS</i>	31

DAFTAR GAMBAR

Gambar	Halaman
1. Mikrokontroler AT89S52	7
2. Blok Diagram Rangkaian	33
3. <i>Flow Chart</i> Program	35
4. Kompilasi Program	38
5. Halaman Utama Aec_Isp_30	39
6. Menu Setup	39

DAFTAR LAMPIRAN

Lampiran	Halaman
1. Listing Program	65
2. Rangkaian Alat Keseluruhan	82

BAB I

PENDAHULUAN

A. Latar Belakang

Zaman sekarang ini manusia selalu menginginkan kemudahan dalam hidupnya, sehingga setiap pekerjaan yang dilakukan tidak menyita waktu yang lama. Perkembangan dunia elektronika dan teknologi pada saat ini yang memberikan kemudahan serta dapat diraih dengan menciptakan berbagai alat yang tersistem secara otomatis. Misalnya, pada umumnya pintu pagar masih dibuka secara manual, sehingga ketika hari hujan maka *user* (pengguna) akan mengalami kesulitan untuk membuka atau menutup pintu pagar, hal ini dirasakan kurang efektif dan efisien.

Proses membuka dan menutup pintu pagar dapat digantikan dengan menggunakan sistem yang lebih canggih, seperti dengan menggunakan *hand phone*, selain untuk berkomunikasi *hand phone* juga dapat digunakan sebagai alat kontrol. Salah satu contoh pengaplikasian *hand phone* adalah sebagai pengendali pintu pagar otomatis, dengan menghubungkan sebuah rangkaian mikrokontroler maka pintu pagar dapat dibuka dan ditutup dari jarak jauh sehingga lebih menghemat waktu dan tenaga.

Sistem ini dirancang dengan pengendali jarak jauh yang dikendalikan melalui pesan singkat atau yang dikenal dengan nama SMS tetapi didalam sistem ini ada beberapa jalur alternatif yang digunakan jika terjadi kendala pada saat membuka atau menutup pintu pagar yaitu penggunaan kode PIN dan kunci manual. Apabila terjadi kendala atau masalah pada saat pengiriman

SMS seperti SMS tidak dapat dikirim dikarenakan gangguan jaringan, sehingga pintu pagar tidak dapat dibuka atau ditutup maka pagar dapat dibuka secara manual dengan memasukan kode PIN sedangkan penggunaan kunci manual digunakan apabila seluruh sistem pembuka pagar mengalami kerusakan seperti power suply rusak maka kunci manual digunakan untuk membuka pagar.

Berdasarkan uraian yang telah dikemukakan perlu dirancang suatu software supaya proses untuk membuka pintu pagar dapat dilakukan menggunakan SMS dan kode PIN untuk alat yang dituangkan dalam bentuk Proyek Akhir dengan judul **“Perancangan dan Pembuatan *Software* Pengendali Pintu Pagar Otomatis Menggunakan SMS dan Kode PIN Berbasis Mikrokontroler AT89S52”**. Sedangkan bagian *hardware* dikerjakan oleh Tomi Andika Putra (2008/03298) dengan judul **“Perancangan dan Pembuatan Alat Pengendali Pintu Pagar Otomatis Menggunakan SMS dan Kode PIN Berbasis Mikrokontroler AT89S52”**.

B. IDENTIFIKASI MASALAH

Berdasarkan latar belakang masalah, dapat diidentifikasi masalah sebagai berikut:

1. Bagaimana merancang kemudahan dalam hidup manusia terutama dalam melakukan pekerjaan.
2. Bagaimana merancang dan membuat *software* pengendali pintu pagar secara otomatis melalui SMS berbasiskan mikrokontroller AT89S52.

3. Bagaimana merancang dan membuat software pengendali pintu pagar menggunakan kode PIN berbasis mikrokontroler AT89S52 dapat ditampilkan ke LCD.
4. Pintu pagar yang dikendalikan secara manual dirasakan masih kurang efektif dan efisien sehingga tidak dapat bekerja lebih optimal.

C. BATASAN MASALAH

Dalam penulisan proposal proyek akhir ini, dibuat suatu batasan masalah yang ada yaitu:

1. Perancangan dan pembuatan *software* pengendali pintu pagar secara otomatis melalui SMS berbasis mikrokontroler AT89S52.
2. Perancangan dan pembuatan *software* sistem otomatisasi pengendali pintu pagar menggunakan kode PIN berbasis mikrokontroler AT89S52.
3. Perancangan *software* untuk proses pengiriman dan penerimaan password dari *keypad* sebagai input data yang ditampilkan pada LCD untuk membuka dan menutup pintu pagar.
4. Bahasa *Assembly* sebagai bahasa pemrograman dan IC AT89S52 yang digunakan untuk membuka dan menutup pintu pagar secara otomatis menggunakan SMS dan kode PIN.

D. Perumusan Masalah

Berdasarkan batasan masalah dapat dirumuskan permasalahannya yaitu

“Bagaimana Merancang dan Membuat *Software* Pengendali Pintu Pagar

Otomatis Menggunakan SMS dan kode PIN Berbasis Mikrokontroler AT89S52”.

E. Tujuan Proyek Akhir

1. Menghasilkan sebuah *software* alat kontrol secara otomatis yang dapat diimplementasikan untuk membuka dan menutup pagar berbasis mikrokontroler AT89S52 menggunakan SMS.
2. Terbentuknya *software* sistem otomatisasi pengendali pintu pagar menggunakan kode PIN berbasis mikrokontroler AT89S52.
3. Terbentuknya *software* untuk proses pengiriman dan penerimaan password dari *keypad* sebagai input data yang ditampilkan pada LCD untuk membuka dan menutup pintu pagar.

F. Manfaat Proyek Akhir

Adapun manfaat dari perancangan dan pembuatan proyek akhir ini adalah sebagai berikut:

1. Memberikan kemudahan dalam mengontrol pagar karena pagar dapat dikontrol dari jarak jauh secara otomatis.
2. Diperolehnya suatu *software* untuk pengendali pintu pagar otomatis menggunakan SMS dan kode PIN berbasis mikrokontroler AT89S52.
3. Memberikan kemudahan dan kenyamanan apabila *software* diterapkan kepada pintu pagar.
4. Memberikan nilai kemajuan khususnya dibidang teknologi karena saat ini kebanyakan pintu pagar pengontrolannya masih secara manual.

BAB II

LANDASAN TEORI

A. Perangkat Lunak (*Software*)

Software atau perangkat lunak adalah program komputer yang merupakan suatu instruksi yang harus diberikan kepada unit pengolah agar komputer dapat menjalankan pekerjaan sesuai dengan yang dikehendaki. Program tersebut ditulis dalam bahasa khusus yang dimengerti oleh mesin.

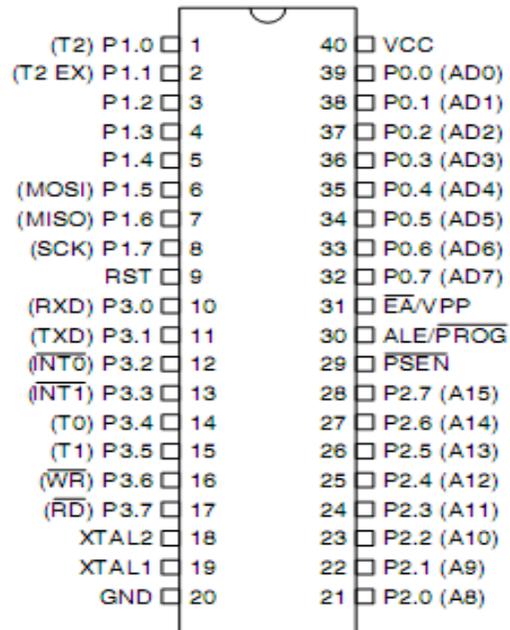
Sommerville (2001: 5) mengemukakan bahwa perangkat lunak tidak hanya mencakup program, tetapi juga semua dokumentasi dan konfigurasi data yang berhubungan, yang diperlukan untuk membuat agar program beroperasi dengan benar. Sistem perangkat lunak biasanya terdiri dari sejumlah program yang terpisah, *file – file* konfigurasi yang digunakan untuk membuat program ini. Dokumentasi sistem yang mendeskripsikan struktur sistem dan dokumentasi *user* yang menjelaskan bagaimana penggunaan sistem tersebut.

B. Mikrokontroler AT89S52

Mikrokontroler AT89S52 merupakan salah satu keluarga dari MSC-51 keluaran Atmel. Jenis mikrokontroler ini pada prinsipnya dapat digunakan untuk mengolah data per bit ataupun data 8 bit secara bersamaan. Sebuah mikrokontroler dapat bekerja bila dalam mikrokontroler tersebut terdapat sebuah program yang berisi instruksi-instruksi yang akan digunakan untuk menjalankan sistem mikrokontroler tersebut. Instruksi-instruksi dari sebuah program pada tiap jenis mikrokontroler mempunyai beberapa perbedaan,

misalnya saja instruksi pada mikrokontroler Atmel berbeda dengan instruksi pada mikrokontroler Motorola. Pada prinsipnya program pada mikrokontroler dijalankan secara bertahap, jadi pada program itu sendiri terdapat beberapa set instruksi dan tiap instruksi itu dijalankan secara bertahap atau berurutan. Menurut Usman (2008: 59) fasilitas yang dimiliki oleh mikrokontroler AT89S52 adalah sebagai berikut:

1. Sebuah *Central Processing Unit* 8 bit.
2. Osilator internal dan rangkaian pewaktu (2 *timer*).
3. RAM internal 256 *byte*.
4. Flash memori 8 *Kbyte* dan fungsi penguncian memori program.
5. Enam buah jalur interupsi (dua buah interupsi eksternal dan tiga buah interupsi internal).
6. Empat buah *programmable port I/O* yang masing-masing terdiri dari delapan buah jalur *I/O*.
7. Sebuah *port* serial dengan kontrol serial *full duplex* UART.
8. Kemampuan untuk melaksanakan operasi aritmatika dan operasi logika.
9. Kecepatan dalam melaksanakan interuksi per siklus mikrodetik pada frekuensi 0 sampai 33 Mhz.



Gambar 1. Mikrokontroler AT89S52
Atmel (1997 : 2)

Susunan pin pada mikrokontroler AT89S52 dapat dilihat pada gambar di atas. Penjelasan untuk masing-masing pin dari mikrokontroler adalah sebagai berikut :

1. Vcc digunakan sebagai catu daya pada pin 40.
2. GND digunakan sebagai *ground* pada pin 20.
3. Port 0 merupakan port paralel 8 bit dua arah. Posisi *Low Significant Bit* (LSB) terletak pada pin 39 dan *Most Significant Bit* (MSB) terletak pada pin 32.
4. Port 1 merupakan port paralel 8 bit dua arah. Posisi LSB terletak pada pin 1 dan MSB terletak pada pin 8.
5. Port 2 merupakan port paralel 8 bit dua arah. Port ini mengirim *byte*

alamat-alamat bila dilakukan pengaksesan memori eksternal. LSB terletak pada pin 21 dan MSB terletak pada pin 28.

6. Port 3 merupakan port paralel 8 bit dua arah. LSB terletak pada pin 10 dan MSB terletak pada pin 17. Port ini mempunyai beberapa fungsi khusus seperti ditunjukkan tabel di bawah ini :

Tabel 1. Pin-pin khusus pada port 3 mikrokontroler AT89S52

Pin -pin pada port 3	Fungsi pengganti
P3.0	RXD (<i>port input serial</i>)
P3.1	TVD (<i>port output serial</i>)
P3.2	INT0 (<i>interrupt eksternal 0</i>)
P3.3	INT1 (<i>interrupt eksternal 1</i>)
P3.4	T0 (<i>input eksternal 0</i>)
P3.5	T1 (<i>input eksternal 1</i>)
P3.6	WR (<i>perintah write pada memori eksternal</i>)
P3.7	RD (<i>perintah read pada memori eksternal</i>)

Atmel (1997: 5)

7. RST (*reset*) pada kondisi *high* akan aktif selama dua siklus.
8. ALE/PROG digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
9. PSEN (*Program Store Enable*) merupakan *store* pembacaan ke memori eksternal.
10. Jika EA/VPP pada kondisi *low* maka mikrokontroler menjalankan instruksi-instruksi yang ada pada memori internal.

11. XTAL 1 sebagai masukan dari rangkaian osilator.

12. XTAL 2 sebagai keluaran dari rangkaian osilator.

C. Interupsi pada Mikrokontroler AT89S52

Interupsi adalah suatu proses bagaimana agar suatu perangkat bisa dilayani secara cepat (*real time*). Perangkat akan mengirimkan sinyal atau bit status ke mikrokontroler sehingga mikrokontroler akan meninggalkan pekerjaan yang sedang dilakukannya untuk menanggapi sinyal atau bit status tersebut. Mikrokontroler akan mengambil subrutin khusus yang telah ditentukan alamat awalnya dan mengerjakan perintah-perintah dalam subrutin tersebut. Setelah selesai, mikrokontroler akan kembali mengerjakan program atau kembali ke alamat program dimana terjadinya proses pemanggilan tadi dan *me-reset* kembali bit-bit tersebut, ini di kemukakan oleh Usman (2008: 353).

D. *Special Function Register* (SFR)

Special Function Register (SFR) adalah memori internal yang memiliki fungsi tertentu. Melalui SFR inilah fasilitas-fasilitas internal 8501 bisa diakses dan dimanfaatkan dengan cara membaca atau menulis SFR tersebut, menurut Usman (2008: 8).

Pada *Special Function Register* terdapat beberapa alamat yang bisa dialamati secara bit dan ada yang tidak bisa dialamati secara bit. Pada *Special Function Register* yang bisa diamati secara bit, alamat pada *digit* keduanya adalah digit 0 atau 8, misalnya 80H, 88H, 90H, 98H, dan F8h. *Special*

Function Register Akumulator adalah salah satunya yang sering dipakai untuk dialamati secara bit dan mempunyai alamat E0H, misalnya A.0, A.1, A.2, A.3, A.4, A.5, A.6, dan A.7. Berikut ini adalah penjelasan secara singkat tentang SFR-SFR beserta fungsinya:

1. *Accumulator*

Accumulator adalah merupakan *register* yang berfungsi untuk menyimpan data sementara. *Register Accumulator* ini sering digunakan dalam proses operasi aritmatika, logika, pengambilan data dan pengiriman data. *Register* ini juga dapat dialamati secara bit.

2. *Register B*

Register B dapat digunakan untuk proses aritmatika dan dapat juga difungsikan sebagai *register* biasa.

3. *Register Port*

Pada *register* ini terdapat 4 buah yaitu *register* port 0, port 2, dan port 3. *Register* port ini digunakan sebagai sarana input/output untuk menyimpan data dari atau ke port untuk masing-masing P0, P1, P2, dan P3.

4. *Register Timer*

Mikrokontroler AT89S52 mempunyai tiga buah 16 bit *timer*, yaitu *Timer* 0 dan *Timer* 1 dibentuk oleh register TH0 dan TL0. *Timer* 1 dibentuk oleh register TH1 dan TL1. Perilaku dari register TH0, TH1, TL0 dan TL1 diatur oleh register TMOD dan register TCON.

5. *Register Control*

Ada beberapa *register* yang berisi bit-bit kontrol dan status untuk sistem interupsi, pencacah atau pewaktu, dan port serial, yaitu register-register IP (*Interrupt priority*), IE (*Interrupt Enable*), TMOD (*TimerMode*), TCON (*Timer Control*), SCON (*Serial Control*), dan PCON (*Power Control*).

- a. *Register* IP digunakan untuk mengaktifkan atau menonaktifkan sarana interupsi. IE.0 sampai IE.6 mengatur masing-masing sumber interupsi, sedangkan IE.7 mengatur interupsi secara keseluruhan. Jika IE.7 bernilai 0 maka sistem interupsi akan non aktif atau keadaan dari IE.0 sampai IE.6 tidak diperhatikan.
- b. *Register* TMOD digunakan untuk mengatur mode kerja dari *Timer 0* dan *Timer 1*. Dengan mengatur mode kerja *timer*, *register* ini dapat digunakan mengatur masing-masing *timer* untuk diatur menjadi timer 16 bit, timer 13 bit, atau timer 8 bit yang dapat diisi ulang secara otomatis. Selain itu, register ini juga dapat mengatur agar proses pencacah *timer* dapat dikendalikan melalui sinyal dari luar mikrokontroler.
- c. *Register* TCON digunakan untuk memulai atau menghentikan proses pencacah timer, mengatur sinyal interupsi dari INT0 atau INT1, serta memantau apakah ada sinyal masuk ke INT0 atau INT1.
- d. *Register* SCON digunakan untuk mengatur perilaku dari UART yang diantaranya memantau proses pengiriman dan penerimaan data seri.
- e. *Register* PCON digunakan untuk pemakaian daya pada IC.

6. *Program Status Word (PSW)*

Register PSW ini berisi informasi status program yang mana masing-masing bit menunjukkan kondisi *Central Processing Unit* setelah operasi dijalankan.

7. *Stack Pointer*

Register Stack Pointer merupakan *register* 8 bit yang terletak pada alamat 81H yang mempunyai fungsi untuk menyimpan alamat data pada saat terjadi *interrupt*.

8. *Data Pointer*

Register data pointer atau DPTR merupakan register 16 bit yang terdiri dari *data pointer high byte (DPH)* dan *data pointer low byte (DPL)*.

9. *Serial Data Buffer*

Serial data buffer terletak pada lokasi 99H yang dibagi menjadi dua *register* yang terpisah, yaitu *transmitter buffer* dan *receiver buffer*. Saat data disalin ke *serial data buffer* maka data sesungguhnya diterima dan diteruskan ke dan dari serial port.

E. Pemrograman AT89S52

Sebuah mikrokontroler dapat menjalankan suatu operasi dengan membaca isi memori program, data yang terbaca akan diartikan sebagai perintah yang harus dikerjakan. Alamat memori program yang harus dibaca ditunjukkan oleh SFR PC (*program counter*). Setelah mengerjakan sebuah

instruksi, PC dinaikkan dan menunjuk ke alamat untuk instruksi selanjutnya. Jika sebuah instruksi menyimpan data yang berbeda ke dalam PC, maka instruksi selanjutnya yang harus dikerjakan oleh mikrokontroler bisa terletak di alamat manapun di memori program, menurut Usman (2008: 121).

Dalam pembuatan program pada mikrokontroler juga terdapat beberapa jenis pengalamatan yang perlu diketahui diantaranya sebagai berikut:

1. Pengalamatan data.

Proses pengalamatan ini terjadi pada sebuah perintah ketika nilai *operand* merupakan alamat dari data yang akan diisikan, dipindahkan, atau proses.

Contoh:

```
Mov P0,A
```

Pada contoh di atas selain mengirim data akumulator ke port 0 juga merupakan perintah untuk pemindahan data dari akumulator ke alamat 80H sehingga dapat juga dituliskan `Mov 80H,A` adalah karena port 0 merupakan salah satu I/O dari mikrokontroler yang beralamat 80H.

2. Pengalamatan langsung.

Proses pengalamatan ini terjadi pada sebuah perintah ketika nilai *operand* merupakan data yang akan diproses, biasanya *operand* tersebut selalu diawali dengan tanda '#' seperti berikut:

```
Mov A,#05
```

Mov A,#Data ; pada bagian atas data telah didefinisikan sebagai bilangan tertentu (data EQU 05) contohnya bil 5.

3. Pengalamatan tak langsung.

Proses pengalamatan ini terjadi pada sebuah perintah ketika salah satu operand merupakan *register* berisikan alamat dari data yang akan dituliskan atau dipindahkan. Pengalamatan jenis ini biasanya digunakan untuk melakukan penulisan, pemindahan atau pembacaan berupa data dalam lokasi memori yang mempunyai urutan beraturan. Jika proses dilakukan dengan pengalamatan langsung jumlah baris yang dibutuhkan akan cukup panjang. Contoh penulisan data 50H hingga 57H.

Listingnya:

```
Mov 50H,#08H  
    51H,#08H  
    52H,#08H  
    53H,#08H  
    54H,#08H  
    55H,#08H  
    56H,#08H  
    57H,#08H
```

Dengan menggunakan sistem pengalamatan tak langsung listing dapat dipersingkat sebagai berikut:

```
Mov R0,#50H
```

Loop :

```
Mov @ RO, #08H
```

```
INC RO
```

```
JCNE RO,#58H,Loop
```

Pada listing ini RO digunakan sebagai register yang menyimpan alamat dari data yang akan dituliskan, dengan melakukan penambahan pada isi RO dan mengulang perintah penulisan data ke alamat yang ditunjukkan RO sehingga *register* menunjukkan nilai 57H+1 atau 58H.

Jadi mode pengalamatan ini dipakai untuk mengakses data yang berada dalam memori, tapi tidak disebutkan secara langsung melainkan dititipkan pada suatu *register*.

4. Pengalamatan Segera

Pengalamatan segera pada sebuah operasi dilakukan dengan memberikan data secara langsung tanpa perantara dari alamat tertentu. Biasanya pengalamatan segera ini diawali dengan tanda #.

Contoh :

```
Mov A, #10H
```

Intruksi ini melakukan operasi memindahkan data 10H kedalam akumulator.

```
Mov A, #20H
```

Intruksi ini melakukan operasi memindahkan data 20H kedalam alamat *register* R1.

5. Pengalamatan kode.

Pengalamatan kode merupakan pengalamatan ketika *operand* merupakan alamat dari instruksi JUMP dan CALL, biasanya *operand* tersebut akan menunjukan ke satu alamat yang telah diberikan label sebelumnya. Contoh sebagai berikut:

A Call delay

Delay

Mov B,#0FFH

Loopdelay:

DJNZ B, Loop delay

Ret.

Pada contoh diatas perintah A CALL delay mempunyai *operand* yang menunjukan ke label sehingga pada saat perintah ini dijalankan program akan melompat ke lokasi memori yang diberi label *delay*.

6. Pengalamatan bit.

Proses pengalamatan dimana ketika *operand* menunjukan ke alamat pada RAM Internal maupun register fungsi khusus yang mempunyai

kemampuan pengalamatan secara bit. Pengalamatan bit terdiri beberapa macam sebagai berikut:

- a. langsung menunjuk kealamat bit.

Perintah ini memberikan logika 1 pada bit dialamat B0H dengan pengalamatan secara bit, contoh:

SetB 0B0H

- b. Menggunakan operator titik.

Perintah ini memberikan logika 1 pada bit ke 0 dari port 3, bit tersebut terletak di alamat B0H dengan pengalamatan secara bit, contoh:

SetB P3.0

F. Bahasa *Assembly*

Mohamed (1984: 354) mengemukakan bahwa assembler merupakan program yang digunakan untuk menterjemahkan kode program bahasa *assembly* yang telah dibuat dengan suatu program editor menjadi kode objek atau kode mesin yang dapat dijalankan oleh mikrokontroler.

1. Simbol-simbol dalam bahasa *assembler*

Program *assembler* juga terdiri atas beberapa simbol yang dijelaskan sebagai berikut:

a. *Label*

Label biasanya dituliskan dengan kata atau istilah tertentu yang mudah dikenali. *Label* mewakili nomor memori program dari instruksi pada baris bersangkutan. Untuk menyatakan nomor memori program yang dituju, label ditulis dalam bagian operand. *Label* berguna untuk memberikan nama pada alamat-alamat yang dituju untuk mempermudah programmer dalam membuat program. Dalam program sebuah *label* harus diawali dengan huruf, tidak boleh ada *label* yang sama dan maksimal 16 karakter.

b. *Mnemonic*

Mnemonic merupakan singkatan dari perintah *mnemonic* bisa juga disebut sebagai kode operasi, yaitu kode-kode yang akan dikerjakan oleh program *assembler* yang ada pada komputer ataupun mikrokontroler. Operasi yang dikerjakan oleh mikrokontroler sangat bergantung pada jenis mikrokontroler yang digunakan, contoh, untuk MCS51 digunakan MOVX, ADD, MOV dan lainnya. Sedangkan kode operasi yang dikerjakan oleh program *assembler* yang ada pada komputer sangat tergantung pada program *assembler* yang digunakan. Contoh :ORG,EQU,DB dan lainnya.

c. *Operand*

Operand merupakan pelengkap dari *mnemonic*. Jumlah *operand* yang dibutuhkan oleh sebuah *mnemonic* tidak selalu sama, sebuah

mnemonic dapat memiliki tiga, dua, satu bahkan tidak memiliki operand seperti terlihat pada contoh berikut:

CJNE R7,#07H,Aksi : memerlukan tiga operand

MOVX @DPTR,A : memerlukan dua operand

SETB A : memerlukan satu operand

NOP : tidak memerlukan operand

d. Komentar

Bagian komentar tidak mutlak ada dalam sebuah program, namun bagian ini seringkali dibutuhkan untuk menjelaskan proses kerja ataupun catatan tertentu pada baris program. Contoh.

MOV A,P1 : Mengambil data dari port 1

Penggunaan komentar biasanya diawali dengan tanda ';' dan dapat diletakan pada bagian manapun dari suatu program.

Beberapa instruksi yang digunakan dalam pembuatan program ini adalah :

1) MOV

MOV berfungsi untuk memindahkan atau menyalin data.

2) ACALL

Instruksi dapat melakukan lompatan ke suatu subrutin yang ditunjukkan oleh alamat pada *label* lompatan yang dilakukan berada diareal 2 K byte.

Contoh: Acall delay' memanggil label delay'

 Delay: 'delay yang dipanggil'

3) RET

Intruksi ini dapat melakukan lompatan untuk kembali melaksanakan instruksi yang dilompatan oleh instruksi *call* atau *acall*.

Contoh: acall delay 'memanggil label delay'

 Instuksi

 Delay: 'delay yang dipanggil'

 Ret 'kembali ke instruksi setelah call'

4) SJMP

Instruksi SJMP (*short jump*) pada dasarnya sama dengan instruksi LJMP, namun yang membedakan adalah ukuran instruksi, dimana pada LJMP memiliki ukuran 3 byte sedangkan SJMP hanya 2 byte. Lokasi memori program dalam instruksi ini tidak dinyatakan dengan lokasi memori program yang sesungguhnya, tapi dinyatakan dengan pergeseran relatif terhadap nilai program counter saat

instruksi ini dilakukan. Pergeseran relatif tersebut dinyatakan dengan 1 *byte* bilangan 2's *complement*, yang biasanya dipakai untuk menyatakan nilai antara - 128 - + 127. nilai minus dipakai untuk menyatakan nilai untuk menyatakan bergeser ke instruksi-instruksi sebelumnya, sedangkan nilai positif digunakan untuk menyatakan pergeseran ke instruksi sesudahnya. Meskipun instruksi SJMP hanya sebatas -128 - +127, tapi instruksi ini tidak dibatasi dengan pengertian daerah memori program 2 K *byte* yang membatasi AJMP (*Absolute jump*).

5) JB (*jump on bit set*)

Instruksi ini melakukan pengujian bit pada alamat bit yang ditunjukkan. Jika data bit =1 maka program tersebut akan melompati ke subrutin yang ditunjukkan oleh instruksi. Akan tetapi bila data bit =0 maka program akan menjalankan instruksi dibawahnya atau sebaliknya.

Contoh: JB P1.0, alamat 1

 Mov A,#05H

Alamat 1:

 Mov R1,#00H

Jika bit tersebut set, program akan lompat ke label alamat 1 dan menjalankan instruksi Mov R1,#00H. namun, jika bit tersebut clear

program akan menjalankan instruksi `Mov A,#05H` terlebih dahulu sebelum menjalankan instruksi di label `alamat1`.

6) `JNB` (*jump on not bit set*)

Instruksi ini melakukan pengujian bit pada alamat yang ditunjuk. Jika data bit bernilai 0 maka program akan melompat ke subrutin yang ditunjuk oleh instruksi. Bila data bit bernilai 1 maka program akan menjalankan instruksi dibawahnya atau selanjutnya.

Contoh: `Hidup : Jnb P3.0, hidup : lompatan kelabel hidup jika port 3.0 berlogika 0`

7) `DJNZ` (*decrement and jump if not zero*)

Instruksi ini melakukan operasi pengurangan pada alamat register serbaguna (`R0..R7`) yang ditunjukkan dengan nilai 1 dan akan jump bila hasil dari pengurangan itu nilainya tidak sama dengan 0. akan tetapi jika hasilnya sama dengan 0 maka program akan menjalankan instruksi dibawahnya.

Contoh :

`LDelay :`

`Djnz R2,LDelay1`

`Ret`

Instruksi `Djnz R2,LDelay 1` akan selalu melakukan lompatan ke label dan mengurangi data pada register 2 dengan 1 selama data dari register 2 belum mencapai nol.

8) SETB dan CLR

Mikrokontroler seri 51 dari Atmel memiliki sebuah prosesor Boolean yang lengkap. Oleh sebab itu mikrokontroler tersebut juga dapat melakukan instruksi Boolean. SETB dan CLR adalah salah satu dari beberapa instruksi Boolean yang dapat dilaksanakan oleh mikrokontroler. CLR adalah intruksi untuk mengubah bit-bit pada RAM Internal ataupun *register* yang dapat diamati secara bit menjadi 0. Sedangkan SETB adalah kebalikan dari CLR, dimana CLR mengubah bit-bit tersebut menjadi 1.

SETB P1.0 'mengubah nilai bit pada P1.0 menjadi 1'

CLR P1.0 'mengubah nilai bit pada P1.0 menjadi 0'

Assembler juga menyediakan simbol-simbol *assembler* khusus yang digunakan untuk mode-mode pengalamatan melalui *register*. Simbol-simbol *assembler* khusus ini mencakup A, R0 sampai dengan R7, DPTR, C, PC dan AB juga tanda dolar (\$) yang dapat digunakan untuk menunjuk nilai pencacah program (*program counter*) saat ini.

2. *Assembler Directive* (pengarah *assembler*)

Menurut Agfianto (2002: 112) *assembler directive* merupakan instruksi pada program *assembler* itu sendiri, bukan sebagai instruksi bahasa *assembler* yang akan dijalankan mikrokontroler yang bersangkutan.

Beberapa pengarah *assembler* yang dikelompokan sebagai berikut :

- a. Kontrol kondisi *assembler*
 - 1) Pengarah *ORG* yang digunakan untuk lokasi awal program.
 - 2) Pengarah *END* merupakan pernyataan yang paling akhir dari program.
 - 3) Pengarah *using* yang memberitahukan *bank register* mana yang aktif.
- b. Defenisi simbol digunakan untuk membuat simbol yang menyatakan *segment*, *register*, bilangan dan alamat. (*segment*, *equ*, *data*, *idata*, *xdata*, *bit*, *code*).
- c. Pemesanan inisialisasi penyimpanan, pengarah ini digunakan untuk menginisialisasi atau memesan tempat penyimpanan baik dalam satu *work*, *byte*, maupun *bit*.

3. *Character string*

String atau karakter yang berupa '*text*' dapat digunakan pada suatu ekspresi, *string* dimaksudkan untuk mempermudah penulisan dan

pemahaman program, sedangkan yang disimpan pada memori adalah kode ASCII dari tiap-tiap karakter, *string* diapit oleh tanda *quotes*(*'*), seperti contoh dibawah ini; Subb a,#'A' ;a = a - 41h.

4. Operator aritmatika

Operasi dasar aritmatika seperti pada operator aritmatika yang digunakan pada penulisan program tingkat tinggi, yaitu;

+ tambah

- kurang

* perkalian

/ pembagian

MOD (*modulo*), atau sisa pembagian

Pada pemrograman *Assembler* juga terdapat instruksi operasi aritmatika yaitu penjumlahan (Add, Addc), pengurangan (Subb), perkalian (Mul) dan pembagian (Div).

Contoh : *Add A, #10H*

A = data pada *register* A + 10H

Subb A, #20H

A = data pada *register* A - 20H - C

Operasi-operasi dasar pada *operand* digunakan untuk mempermudah penulisan dan pemahaman program, sedangkan setelah dikompilasi, hasil dari operasi tersebut yang akan ditulis. Contoh dalam penulisan program;

MOV A,#20 MOD 3 ;A diisi dengan 2 (sisa pembagian 20/3)

5. Operasi logika

Operasi logika juga seperti pada operasi logika yang digunakan pada penulisan program tingkat tinggi, yaitu;

OR logika OR (atau)

AND logika AND (dan)

XOR logika XOR

NOT logika NOT (bukan)

Contoh; Mov a,#(NOT empat)+1 a=komplemen 4 +1

6. Operator spesial

Operasi ini terdiri dari :

a. SHR (Untuk menggeser kekanan)

Contoh : *Mov A, #00001000b SHR 1* adalah sama dengan *Mov A, #00000100b.*

b. SHL (Untuk menggeser kekiri)

Contoh : *Mov A, #00001000b SHL 1* adalah sama dengan *Mov*

A,

#00010000b.

c. **HIGH** (Untuk mengambil nilai *byte* tinggi)

Contoh : *Mov A, #HIGH 1234H* adalah sama dengan *Mov A,*

#12H.

d. **LOW** (Untuk mengambil nilai *byte* rendah)

Contoh : *Mov A, #LOW 1234H* adalah sama dengan *Mov A,*

#34H.

e. **()** (Untuk operasi yang harus didahulukan)

Contoh : *Mov A, #(10 + 4) * 3*, bilangan 10 *decimal* terlebih dahulu dijumlahkan dengan 4 sebelum dikali dengan 3 dengan adanya operator **()**.

G. Teknologi SMS (*Short Message Service*)

Penggunaan SMS sudah dimulai sejak tahun 1991, dan telah dipakai secara luas di Eropa dan Asia, tetapi di Amerika bagian utara baru tersebar luas penggunaannya setelah itu. Menurut Palmone (2005: <http://kb.palmone.com/>).

Menurut Gupta (2005 : [http:// www.wirelessdevnet.com/](http://www.wirelessdevnet.com/))SMS merupakan salah satu mekanisme pengiriman pesan pendek melalui jaringan HP (*Hand Phone*). SMS disimpan dan dikirim melalui transmisi pesan ke dan

dari jaringan HP. Pesan text yang berasal dari HP pengirim disimpan pada pusat layanan SMS atau *Short Message Service Center* (SMSC) yang kemudian meneruskannya ke HP penerima. Ini berarti bahwa jika HP penerima berada dalam kondisi tidak siap menerima, SMS tersebut akan disimpan dan kemudian dapat dikirimkan. Setiap satu SMS tidak lebih dari 160 karakter. Karakter ini dapat berupa teks (*alphanumeric*) atau biner pesan non-teks.

Elemen utama dalam jaringan SMS adalah SMSC yang di dalamnya terdapat berbagai proses pengolahan SMS. Prinsip kerja sebuah SMSC adalah *store and forward*. Dengan prinsip ini, seluruh SMS yang masuk akan langsung ditampung tanpa melihat status tujuan apakah ada atau tidak. Penyampaian ke tujuan akan dilakukan kemudian dengan terlebih dahulu mengidentifikasi tujuan dengan meng-*query* entitas-entitas yang terlibat. Rozidi (2004: 9-10).

SMS dikirimkan dalam bentuk satuan paket data atau *Packet Data Unit* (PDU) yang dipertukarkan pada lapisan aplikasi dalam protokol *Short Message Peer to Peer Protocol* (SMPP) Rozidi (2004: 14). PDU merupakan sejenis kompresi data atau pemampatan data yang menyertakan informasi nomor pengirim, nomor penerima, nomor SMS *center*, tanggal dan jam selain pesan pokok. Format PDU harus dikembalikan lagi menjadi format *text* yang bisa dibaca sesuai pesan yang dikirimkan ketika PDU sampai pada penerima. Berikut ini adalah struktur dari format PDU sebagaimana dikutip dari Pettersson (<http://www.dreamfabric.com/sms/>) berdasarkan standard *European*

Telecommunication Standards Institute (ETSI), bahwa terdapat dua format PDU yaitu:

1. Format PDU untuk Penerimaan SMS

Contoh PDU:

```
07917283010010F5040BC87238880900F10000993092516195800AE832  
9BFD4697D9EC37
```

Berikut adalah uraian dari rangkaian PDU tersebut:

07 : Informasi panjang atau pasangan digit nomor SMSC

91 : Jenis nomor SMSC. Angka 91 menandakan format nomor internasional

7283010010F5 : Nomor SMSC yang digunakan. Setelah diterjemahkan

nomor SMSC tersebut adalah "+27381000015"

00 : Pengenal protokol (TP-PID)

00 : Skema pengkodean data, dalam hal ini 0

99309251619580 : Penunjuk waktu (semi oktet)

0A : Panjang pesan

E8329BFD4697D9EC37 : Isi pesan "hello"

Semua kode tersebut berupa bilangan heksadesimal 8 bit, kecuali nomor SMSC, nomor pengirim, dan penunjuk waktu yang berupa bilangan desimal semioktet.

2. Format PDU untuk Pengiriman SMS

0011000B916407281553F80000AA0AE8329BFD4697D9EC37

Berikut adalah uraian dari rangkaian PDU tersebut:

00 : Panjang nomor SMSC

11 : Oktet pertama pesan SMS SUBMIT

00 : *Reference* pesan. Nilai 00 berarti membiarkan HP
merefereksi nomornya sendiri

0B : Panjang nomor tujuan

91 : Jenis nomor SMSC. Angka 91 menandakan format nomor
internasional

6407281553F8: Nomor HP tujuan dalam semioktet

00 : Pengenal protokol

00 : Skema pengkodean SMS

AA : Jangka validitas SMS

0A : Panjang SMS

E8329BFD4697D9EC37: Isi pesan "hello"

Format PDU SMS tersebut dapat dibuat atau diterjemahkan dengan menggunakan program PDU converter, seperti PDUSpy yang dapat didownload secara gratis di internet.

H. *AT Command*

Beberapa jenis HP seperti Nokia, Siemens, dan Sony Ericsson memiliki *Modulation-Demodulation* (modem) seperti juga komputer yang memiliki modem untuk mengirim data melalui jaringan telepon. HP yang memiliki modem akan dapat dikoneksikan dengan komputer melalui kabel data sehingga dapat berfungsi sebagai pengirim data seperti *fax*. Untuk digunakan bersama mikrokontroler, HP juga dikoneksikan melalui kabel data.

Perintah standar modem dikenal dengan sebutan *AT Command*. Perintah ini dapat digunakan untuk mengirim, menerima atau membaca, menghapus SMS dan masih banyak lagi fungsinya. Beberapa jenis *AT Command* yang berhubungan dengan penanganan SMS dapat dilihat pada tabel berikut:

Tabel 2: Contoh *AT Command* pada HP Siemens

<i>AT Command</i>	Fungsi
AT + CMGS	Mengirim Pesan
AT + CMGR	Membaca Pesan
AT + CMGF	Format Pesan
AT + CMGD	Menghapus Pesan
AT + CNMI	Prosedur Indikasi Pesan Baru
AT + CPMS	Pemilihan Target Memori
AT + CSMS	Pemilihan Layanan Pesan

Setiap selesai menerima *AT Command*, HP akan mengirimkan respon. Salah satu contoh respon pada HP akibat penerapan *AT Command* seperti berikut ini:

Pengiriman SMS:

```
AT + CMGS = 24 <cr>
```

```
<pdu> <ctrl-z>
```

Maka HP akan memberikan respon:

```
>
```

```
+ CMGS:
```

```
OK
```

Salah satu penerapan *AT Command* diatas yaitu untuk mengirim SMS, perintah AT + CMGS diikuti panjang pasangan karakter dari PDU SMS yang akan dikirim. HP akan memberikan respon berupa karakter > ke *interface* yang mengirimkan perintah tersebut setelah tombol enter ditekan (jika melakukan koneksi dengan komputer) atau setelah kode ASCII enter dikirim (jika melakukan koneksi dengan mikrokontroler). Kemudian HP akan menunggu sampai menerima kode PDU SMS dan diakhiri dengan menekan tombol ctrl-z atau setelah menerima kode ASCII LF (terjemahan dari ctrl-z). Jika pengiriman SMS berhasil, maka akan diberi respon berupa +CMGS diikuti dengan respon OK.

(298283). Dan jika ingin menutup pagar cukup dengan menekan tombol “#”.

LCD yang terdapat pada alat menunjukkan kondisi/status dari peralatan yang dikendalikan, jika alat dalam keadaan terbuka maka akan tampil pada LCD ‘PINTU PAGAR TERBUKA’ dan sebaliknya jika alat dalam keadaan tertutup maka akan tampil pada LCD ‘PINTU PAGAR TERTUTUP’. Apabila pintu pagar macet maka akan tampil pada LCD ‘PINTU PAGAR MACET’. Setiap kegiatan yang dilakukan dengan menggunakan LCD, HP *slave* akan mengirimkan SMS ke HP master berupa status pintu pagar.

BAB V

PENUTUP

A. Kesimpulan

Setelah melakukan tahap perancangan, pembuatan dan proses pengujian serta analisa terhadap *software*, ada beberapa hal yang dapat disimpulkan antara lain:

1. Program pengendali pintu pagar otomatis menggunakan SMS dan kode PIN berbasis mikrokontroler AT89S52 ini dapat diaplikasikan di pintu pagar rumah.
2. Sistem pengendali pintu pagar otomatis menggunakan SMS dan kode PIN berbasis mikrokontroler AT89S52 yang dirancang dan diimplementasikan menggunakan bahasa pemrograman *Assembly*.
3. Alat ini dapat digunakan untuk membuka dan menutup pintu pagar dari jarak jauh dengan menggunakan SMS.
4. Selain untuk membuka dan menutup pintu pagar dari jarak jauh, alat ini juga dapat digunakan untuk mengecek kondisi pintu pagar dari jarak jauh apakah dalam keadaan terbuka, tertutup atau macet.
5. Alat ini hanya dapat menerima perintah nomor telepon seluler yang terprogram dan dengan format yang sudah ditetapkan, selain dari itu alat tidak akan merespon perintah yang dikirimkan.

B. Saran

Beberapa hal yang perlu dipertimbangkan untuk pengembangan *software* pengendali pintu pagar otomatis menggunakan SMS dan kode PIN berbasis mikrokontroler AT89S52 antara lain:

1. Sistem pengendali pintu pagar otomatis menggunakan SMS dan kode PIN berbasis mikrokontroler AT89S52 yang dirancang dan diimplementasikan menggunakan bahasa pemrograman *Assembly* ini, diharapkan untuk kedepan dapat dikembangkan dan disesuaikan dengan perkembangan perangkat lunak.
2. Untuk kesempurnaan sistem ini dimasa yang akan datang diharapkan kepada pembaca dan pengguna sistem ini untuk melakukan pengembangan-pengembangan seperti pengembangan sistem ini untuk pintu pagar yang sebenarnya.
3. Sistem yang dibuat masih memiliki kelemahan baik dari alat maupun program, diharapkan nantinya agar dapat diimplementasikan pada sistem pintu pagar.

DAFTAR PUSTAKA

- Atmel. (1997). *Flas Microcontroller: Architectural Overview*. Atmel Inc. <http://atmel.com>. (didownload 18 Oktober 2010)
- Gupta, Puneet. (2005). *Short Message Service: What, How and Where?*. <http://www.wirelessdevnet.com/channels/sms/features/sms.html>. (didownload 16 April 2007)
- Palmone. (2005). *How does SMS work?*. <http://kb.palmone.com/srvs/cgi-bin/webcgi.exe?new>. (didownload 2 November 2010)
- Pettersson, Lars. *SMS and the PDU format*. <http://www.dreamfabric.com/sms/> (didownload 28 Mei 2007)
- Putra, Agfianto Eko. (2002). *Belajar Mikrokontroler AT89C51/52/55*. Yogyakarta: Gava Media
- Rafiqzaman, Mohamed. (1984). *Microprocessors And Microcomputer Development System*. New York: Harper & Row
- Rozidi, Romzi Imran. (2004). *Membuat Sendiri SMS Gateway Berbasis Protokol SMPP*. Yogyakarta: Andi
- Sommerville, Ian. (2001). *Software Engineering Rekayasa Perangkat Lunak Edisi 6 Jilid 1*. Jakarta : Erlangga
- Usman. (2008). *Teknik Antarmuka dan Pemrograman Mikrokontroler AT89S52*. Yogyakarta: Andi