

**RANCANG BANGUN SIMULASI PENSINYALAN
KOMUNIKASI SELULER BAGIAN *SUBSCRIBER* PENGIRIM
MENGUNAKAN MIKROKONTROLLER ATMEL 89S51**

SKRIPSI

OLEH

YANI BARLIANI DWIANNA

04 04 03 085 7



**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

**RANCANG BANGUN SIMULASI PENSINYALAN
KOMUNIKASI SELULER BAGIAN *SUBSCRIBER* PENGIRIM
MENGUNAKAN MIKROKONTROLLER ATMEL 89S51**

SKRIPSI

OLEH

YANI BARLIANI DWIANNA

04 04 03 085 7



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/ 2008**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

**RANCANG BANGUN SIMULASI PENSINYALAN KOMUNIKASI
SELULER BAGIAN *SUBSCRIBER* PENGIRIM MENGGUNAKAN
MIKROKONTROLLER ATMEL 89S51**

yang dibuat untuk melengkapi sebagian persyaratan sebagai Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang merupakan sumber informasinya dicantumkan sebagaimana mestinya .

Depok, 14 Juli 2008

Yani Barliani Dwianna

NPM 04 04 03 085 7

PERSETUJUAN

Skripsi dengan judul:

**RANCANG BANGUN PENSINYALAN KOMUNIKASI SELULER
BAGIAN *SUBSCRIBER* PENGIRIM MENGGUNAKAN
MIKROKONTROLLER ATMEL 89S51**

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. dan disetujui untuk diajukan dalam sidang ujian skripsi.

Depok, 14 Juli 2008

Dosen Pembimbing

Abdul Muis, S.T., M.Eng., Ph.D

NIP. 132 233 210

UCAPAN TERIMA KASIH

Puji syukur kehadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

Abdul Muis, S.T., M.Eng., Ph.D

Dr. Ir. Arman Djohan Diponegoro, M.Eng

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan saran, diskusi, pengarahan, dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.

Yani Barliani Dwianna
NPM 04 04 03 085 7
Departemen Teknik Elektro

Dosen Pembimbing
Abdul Muis, S.T., M.Eng., Ph.D

**RANCANG BANGUN PENSINYALAN KOMUNIKASI SELULER
BAGIAN *SUBSCRIBER* PENGIRIM MENGGUNAKAN
MIKROKONTROLLER ATMEL 89S51**

ABSTRAK

Pada skripsi ini dibuat sebuah rancang bangun simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim dengan menggunakan mikrokontroller Atmel 89S51 yang dapat digunakan untuk mensimulasikan bit-bit yang ada selama proses panggilan keluar antara *mobile station* dengan *Mobile Switching Center*. Bit-bit yang digunakan sudah sesuai dengan bit-bit yang ada pada pensinyalan komunikasi seluler di Indonesia karena didapatkan langsung dari salah satu *vendor* telekomunikasi di Indonesia. Alasan penggunaan mikrokontroller Atmel 89S51 dikarenakan prosesor yang memiliki fitur yang dibutuhkan dan mudah ditemui di pasaran ini mendukung penggunaan bahasa *assembly*. Sehingga jalannya simulasi dapat lebih optimal.

Perancangan simulator pensinyalan ini dilakukan dengan menggunakan perangkat lunak simulasi 8051 IDE. Sedangkan untuk rancang bangunnya dilakukan dengan menggunakan perangkat keras mikrokontroller Atmel 89S51. Perangkat lunak 8051 IDE digunakan untuk menerjemahkan kode *mnemonic* program sehingga dapat dikenali oleh mikrokontroller Atmel 89S51. Program pensinyalan dibuat berdasarkan algoritma pensinyalan yang sesuai dengan *message flow* pensinyalan komunikasi seluler yang umum digunakan di Indonesia. Sedangkan perangkat keras mikrokontroller Atmel 89S51 digunakan untuk membuat rangkaian pensinyalan komunikasi seluler yang terdiri dari dua mikrokontroller, satu *keypad*, dan rangkaian *seven segment*.

Pengujian performa mikrokontroller digunakan untuk melihat kinerja tiap-tiap komponen rangkaian mikrokontroller. Analisis yang dilakukan meliputi cara kerja setiap komponen dan unjuk kerja rangkaian mikrokontroller secara keseluruhan. Dari hasil simulasi dan pengujian dapat disimpulkan bahwa masing-masing sub-sistem rangkaian simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim telah menunjukkan performa yang cukup baik walaupun masih terdapat beberapa kekurangan yang harus disempurnakan. Sehingga secara umum, rangkaian simulator pensinyalan komunikasi seluler yang dibuat sudah cukup merepresentasikan sistem pensinyalan komunikasi seluler bagian *subscriber*.

Kata kunci : Komunikasi Seluler, Pensinyalan, Mikrokontroller Atmel 89S51, 8051 IDE, Assembly

Yani Barliani Dwianna
NPM 04 04 03 085 7
Electrical Engineering Departement

Counselor
Abdul Muis, S.T., M.Eng., Ph.D

**DESIGNING SIGNALLING ON MOBILE CELLULAR
COMMUNICATION ON TRANSMITTER SUBSCRIBER DIVISION
WITH MICROCONTROLLER AT89S51 APLLICATION**

ABSTRACT

This paper realize signalling on mobile cellular communication design on transmitter subscriber division with microcontroller AT89S51 application which can simulate bits through calling process that happens between mobile station and mobile switching center. These bits have been adjusted with bits which are used in mobile cellular communication signalling at one of telecommunication vendor in indonesia. A microcontroller AT89S51 is considered since this processor has capable features, easy to find in general market, and could support assembly as language program. So, the simulation can run optimally.

This signalling in simulator are designed through 8051IDE software simulation. On the other hand, the simulator circuit is designed based on microcontroller Atmel 89S51 hardware. The 8051IDE software is use to translate program mnemonic code, to be used by microcontroller AT89S51. Signalling program made in this paper is based on algorithm program which has been adjusted with the message flow of mobile cellular communication in indonesia. While the sub-systems of microcontroller AT89S51 hardware consists of two microcontroller AT89S51 minimum circuit, one keypad circuit, and one seven segment circuit.

Test was given to evaluate the sub-system and the simulator main circuit performance. Analysis were made based on each sub system and the whole of main simulator circuit. The experimental results show that the performance of each sub system and the whole main simulator circuit produce good results and accurate enough although developments are still needed. As general, this mobile cellular communication signalling simulator may represent mobile cellular communication signalling system in indonesia.

Keywords : Cellular Communication, Signalling, Microcontroller Atmel 89S51, 8051 IDE, Assembly

DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN SKRIPSI	ii
PERSETUJUAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
DAFTAR SINGKATAN	xiv
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH	2
1.3 TUJUAN PENELITIAN	2
1.4 BATASAN MASALAH	3
1.5 METODOLOGI PENELITIAN	3
1.6 SISTEMATIKA PENULISAN	3
BAB II DASAR TEORI	5
2.1 PENSINYALAN PADA KOMUNIKASI SELULER	5
2.1.1 Pensinyalan	6

2.1.1.1	<i>Kondisi Idle</i>	9
2.1.1.2	<i>Panggilan</i>	11
2.1.1.2.1	<i>Panggilan Keluar</i>	11
2.1.1.2.2	<i>Panggilan Masuk</i>	12
2.1.2	<i>Fungsi-fungsi Layer 3 Pensinyalan</i>	16
2.1.2.1	<i>Connection Management (CM)</i>	17
2.1.2.2	<i>Mobility Management (MM)</i>	17
2.1.2.3	<i>Radio Resource Management (RR)</i>	17
2.2	MIKROKONTROLLER ATMEL 89S51 (AT89S51)	18
2.2.1	<i>Deskripsi Sistem</i>	18
2.2.2	<i>Deskripsi Pin</i>	19
2.3	BAHASA PEMROGRAMAN ASSEMBLY UNTUK AT89S51	22
2.3.1	<i>Program Sumber Assembly</i>	23
2.3.1.1	<i>Bagian Label</i>	23
2.3.1.2	<i>Bagian Kode Operasi</i>	23
2.3.1.3	<i>Bagian Operand</i>	24
2.3.1.4	<i>Bagian Komentar</i>	24
BAB III		26
	RANCANG BANGUN RANGKAIAN SIMULATOR PENSINYALAN KOMUNIKASI SELULER DENGAN MIKROKONTROLLER AT89S51	26
3.1	KONSEP RANCANG BANGUN RANGKAIAN SIMULATOR PENSINYALAN KOMUNIKASI SELULER	26
3.1.1	<i>Program-program Pensinyalan</i>	27
3.1.1.1	<i>Flow Chart Pensinyalan</i>	28
3.1.1.2	<i>Algoritma Pensinyalan</i>	32
3.1.1.3	<i>Listing Program Pensinyalan</i>	34
3.1.1.3.1	<i>IMSI Attach</i>	34
3.1.1.3.2	<i>Identifikasi Mobile Equipment</i>	35
3.1.1.3.3	<i>Update Lokasi</i>	36
3.1.1.3.4	<i>Proses panggilan keluar pada MS</i>	37
3.1.1.3.5	<i>Program Sentral</i>	39
3.1.1.3.6	<i>Program Delay</i>	39
3.1.2	<i>Rangkaian Mikrokontroler untuk Simulator Pensinyalan</i>	40
3.1.2.1	<i>Sistem Minimum AT89S51</i>	40
3.1.2.2	<i>Rangkaian Keypad</i>	42
3.1.2.3	<i>Rangkaian seven segment</i>	44
3.1.2.4	<i>Proses Burning</i>	46
BAB IV	HASIL UJI COBA DAN ANALISIS RANGKAIAN	49

4.1 HASIL UJI COBA SUB-SISTEM	49
4.1.1 Sub-sistem Rangkaian Sistem Minimum AT89S51	50
4.1.1.1 Rangkaian system minimum untuk program pensinyalan utama	51
4.1.1.2 Rangkaian system minimum untuk program keypad	54
4.1.2 Sub-sistem Rangkaian Keypad	56
4.1.3 Sub-sistem Rangkaian Seven Segment	59
4.2 HASIL UJI COBA SISTEM KESELURUHAN	62
4.2.2 Uji Coba Software 8051IDE	62
4.2.3 Uji Coba Hardware Rangkaian Simulator Sistem Pensinyalan	68
4.3 ANALISA KONDISI PROSES PENSINYALAN PADA RANGKAIAN SIMULATOR SISTEM PENSINYALAN KOMUNIKASI SELULER BAGIAN SUBSCRIBER PENGIRIM	72
4.3.1 Kondisi IMSI Attach	72
4.3.2 Kondisi Identifikasi IMEI	74
4.3.3 Kondisi Perbaruan Lokasi	74
4.3.4 Kondisi Panggilan Keluar	74
BAB V KESIMPULAN	76
DAFTAR ACUAN	77
DAFTAR PUSTAKA	78
LAMPIRAN	79

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Fungsi-fungsi pada Komunikasi Seluler.	5
Gambar 2.2. Bagan Sederhana Pensinyalan Komunikasi Seluler.	6
Gambar 2.3. Blok Diagram Mobile Station [6]	7
Gambar 2.4. Aliran Pesan saat melakukan Identifikasi ME [5]	9
Gambar 2.5. Aliran Pesan saat Perbaruan Lokasi [5]	10
Gambar 2.6. Aliran Pesan Panggilan Keluar [5]	12
Gambar 2.7. Aliran Pesan Panggilan Masuk	14
Gambar 2.8. Message Flow saat MS tujuan Mati [6]	14
Gambar 2.9. Message Flow saat MS mengaktifkan CFN [6]	15
Gambar 2.10. Message Flow saat MS Menolak Semua Panggilan Masuk [6]	15
Gambar 2.11. Message Flow saat MS tidak diketahui di HLR [6]	15
Gambar 2.12. Fungsi-fungsi protokol layer 3 pada pensinyalan komunikasi seluler	16
Gambar 2.13. Blok Diagram AT89S51 [11]	19
Gambar 2.14. Konfigurasi Pin AT89S51 [11]	22
Gambar 3.1 Flow chart pensinyalan kondisi idle	29
Gambar 3.2 Flow chart pensinyalan proses komunikasi utuh pada subscriber	31
Gambar 3.3 Blok diagram chip dan port sistem minimum mikrokontroler AT89S51	32
Gambar 3.4 Rangkaian Sistem Minimum AT89S51 [11]	41
Gambar 3.5 Rangkaian Keypad	42
Gambar 3.6 Skema Konfigurasi Pin Keypad	43
Gambar 3.7 Rangkaian Seven Segment	45
Gambar 3.8 Connection Diagram BCD to 7-segment Decoder 74LS47N	46
Gambar 3.9 Tahap Inisialisasi	47
Gambar 3.10 Tahap Load Program	47
Gambar 3.11 Tahap Burning	48
Gambar 4.1 Rangkaian simulator pensinyalan komunikasi seluler bagian subscriber pengirim secara keseluruhan.	49

Gambar 4.2 dua rangkaian system minimum yang telah terhubung	50
Gambar 4.3 rangkaian system minimum untuk program pensinyalan utama	51
Gambar 4.4 rangkaian system minimum untuk program keypad	54
Gambar 4.5 sub system rangkaian keypad	57
Gambar 4.6 sub system rangkaian seven segment	60
Gambar 4.7 Tampilan output saat terjadi kesalahan	63
Gambar 4.8 Tampilan letak dan jenis kesalahan program	63
Gambar 4.9 Tampilan letak dan jenis kesalahan program	64
Gambar 4.10 Tampilan register	64
Gambar 4.11 Tampilan internal memori	67
Gambar 4.12 Tampilan Port rangkaian utama	68
Gambar 4.13 Rangkaian simulator keseluruhan sebelum di-burn	69
Gambar 4.14 Rangkaian simulator keseluruhan setelah di-burn	70

DAFTAR TABEL

	Halaman
Tabel 2.1 Fitur-fitur khusus pada <i>port</i> 3 [11]	21
Tabel 2.2 Bentuk program <i>assembly</i> umum [11]	24
Tabel 3.1. Tabel Definisi Alamat pada <i>SIM Card</i>	33
Tabel 3.2. Tabel Definisi Alamat pada Memori Internal	33
Tabel 3.3. Tabel representasi bit <i>port</i> 2 saat tombol <i>keypad</i> ditekan	43
Tabel 4.1. Tabel bit-bit <i>message type</i> dari <i>message flow</i> pensinyalan	59
Tabel 4.2. Nilai yang tidak ditampilkan	65
Tabel 4.3. Nilai yang ditampilkan	65
Tabel 4.4. perbedaan fungsi <i>port</i> pada rangkaian <i>keypad</i> dengan rangkaian utama	67

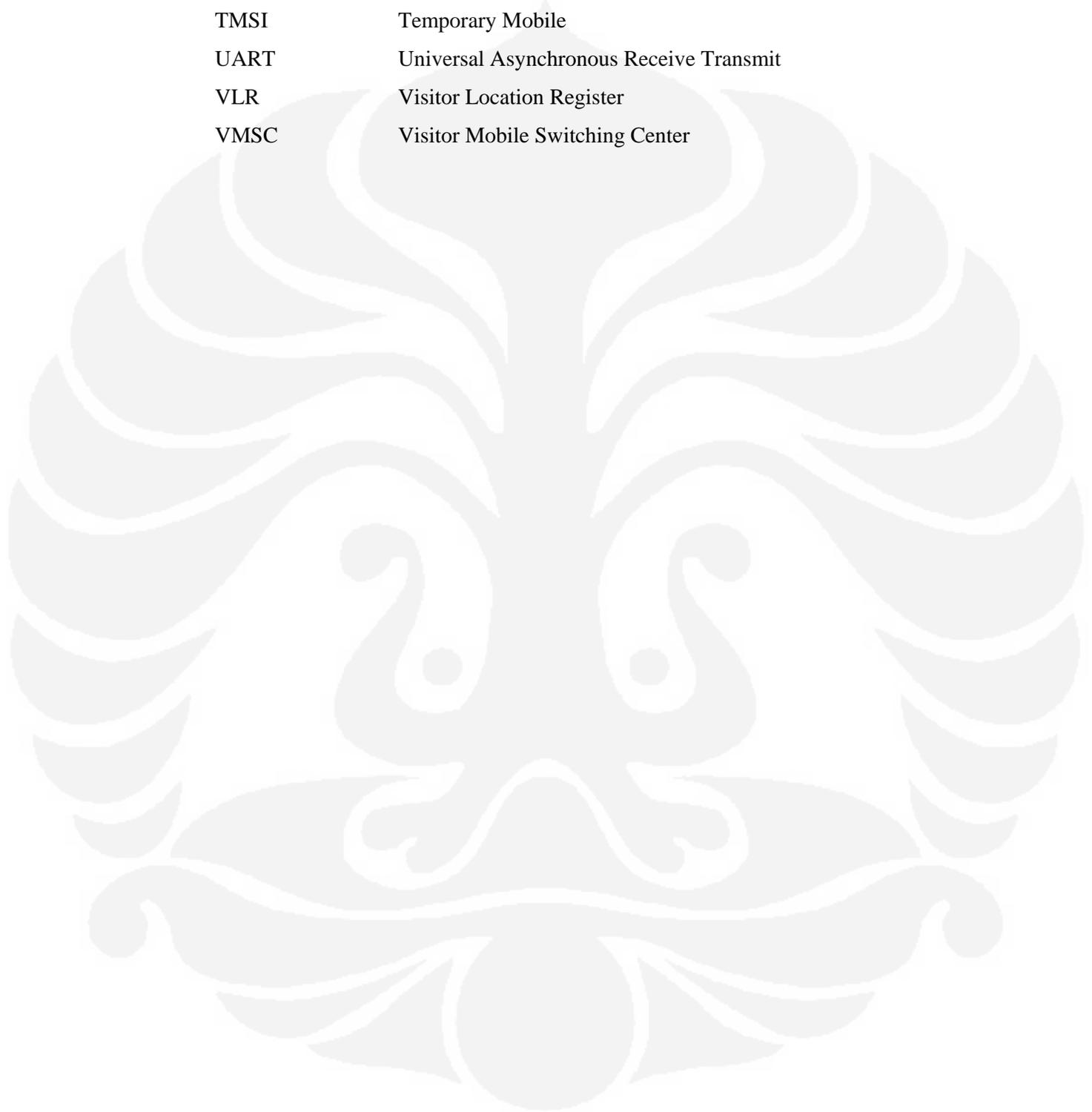
DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Listing Program Prosedur IMSI Attach	79
Lampiran 2 Listing Program Prosedur Identifikasi ME	80
Lampiran 3 Listing Program Prosedur Perbaruan Lokasi	81
Lampiran 4 Listing Program Prosedur Panggilan Keluar	82
Lampiran 5 Listing Program Prosedur Delay	83
Lampiran 6 Listing Program Sentral	83
Lampiran 7 Listing Program Deklarasi Variabel dan Deklarasi Alamat	85
Lampiran 8 Listing Program Nilai Pada Sentral	87
Lampiran 9 Listing Program Menampilkan Ke Seven Segment	87
Lampiran 10 Listing Program Keypad	88

DAFTAR SINGKATAN

BCCH	Broadcast Control Channel
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CC	Call Control
CFB	Call Forwarding Busy
CFNRc	Call Forwarding Not Reachable
CFNRy	Call Forwarding Not Replay
CFU	Call Forwarding Unconditional
CM	Call Management
EEPROM	Electrically Erasable Programmable Read Only Memory
GMSC	Gateway Mobile Station Center
HLR	Home Location Register
IMSI	International Mobile
ISP	In System Programming
LA	Location Area
ME	Mobile Equipment
MM	Mobile Management
MS	Mobile Station
MSC	Mobile Switching Center
MSISDN	Mobile Subscriber Integrated Services Digital Network
MSRN	Mobile Station Roaming Number
NSS	Network and Switching Subsystem
OSS	Operation Support Subsystem
PLMN	Public Land Mobile Network
PSTN	Public Switched Telephone Network
RAM	Random Access Memory
ROM	Read Only Memory
RR	Radio Resources Management

SMS	Short Message Service
SS	Supplementary Service
TMSI	Temporary Mobile
UART	Universal Asynchronous Receive Transmit
VLR	Visitor Location Register
VMSC	Visitor Mobile Switching Center



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Komunikasi selular merupakan salah satu teknologi yang sangat memasyarakat di era modern dengan perkembangan teknologi yang sangat pesat. Hampir setiap orang diberbagai kalangan masyarakat akrab dengan komunikasi selular. Ponsel telah menjadi barang yang sangat dikenal masyarakat saat ini khususnya masyarakat perkotaan. Indonesia sebagai negara padat penduduk tentunya juga merupakan negara dengan pengguna komunikasi selular dengan jumlah yang tidak sedikit. Namun sangat disayangkan hingga saat ini indonesia belum mampu membuat produk sendiri dari teknologi komunikasi selular tersebut.

Produk dari komunikasi seluler sangat beragam. Hal ini ditandai dengan perkembangan komunikasi seluler itu sendiri yang dimulai dari *first generation*, yaitu komunikasi seluler yang menggunakan teknologi analog. Perkembangan berikutnya dari *first generation* adalah *second generation* yaitu komunikasi seluler yang menggunakan teknologi digital. Hingga saat ini, perkembangan komunikasi seluler telah mencapai tahap *third generation* bahkan sudah mulai didengung-dengungkan mengenai *fourth generation*. Namun fenomena yang terjadi di masyarakat khususnya di indonesia, masyarakat hanya bertindak sebagai konsumen tanpa pernah berpikir untuk membuat sendiri produk dari komunikasi seluler tersebut. Hal ini disebabkan salah satunya karena minimnya pengetahuan masyarakat tentang teknologi komunikasi seluler itu sendiri.

Pada komunikasi seluler terdapat sebuah fungsi penting yang memungkinkan terjadinya sebuah hubungan komunikasi yang merupakan tujuan dari diciptakannya komunikasi seluler itu sendiri. Fungsi tersebut adalah pensinyalan. Pensinyalan merupakan sebuah mekanisme yang menghubungkan *mobile station* atau yang lebih dikenal sebagai ponsel dengan sebuah sentral. Dengan mekanisme ini, maka ponsel dapat terus terhubung sehingga komunikasi dapat terjadi dimana saja pengguna komunikasi seluler berada.

Pada skripsi ini dibuat rancang bangun pensinyalan komunikasi seluler bagian *subscriber* pengirim dengan menggunakan mikrokontroler Atmel 89S51. Rangkaian mikrokontroler ini diharapkan dapat mensimulasikan skenario pensinyalan beserta tipe *message* dan bit-bit yang terkandung di dalamnya. Sebelum membuat rangkaian mikrokontroler pensinyalan, terlebih dahulu dibuat program-program dasar yang diperlukan selama proses pensinyalan berlangsung, baik pada *mobile station* maupun pada *mobile switching center*. Dalam membuat program dasar pensinyalan, bit-bit yang digunakan telah disesuaikan dengan bit-bit yang digunakan pada pensinyalan komunikasi seluler saat ini di Indonesia. Diharapkan dengan adanya kesesuaian ini, hasil rangkaian simulasi pensinyalan komunikasi seluler yang dibuat benar-benar dapat merepresentasikan pensinyalan komunikasi seluler yang ada saat ini.

1.2 PERUMUSAN MASALAH

Masalah yang akan diteliti pada skripsi ini adalah bagaimana merancang bangun proses pensinyalan pada komunikasi seluler dengan menggunakan mikrokontroler Atmel 89S51 serta unjuk kerja dari mikrokontroler tersebut. Alasan merancang bangun menggunakan mikrokontroler Atmel 89S51 dikarenakan mikrokontroler ini membutuhkan konsumsi daya yang rendah dan mendukung penggunaan bahasa pemrograman *assembly*. Sehingga dengan menggunakan mikrokontroler jenis Atmel 89S51 diharapkan dapat menghasilkan simulasi dengan kinerja yang cukup baik bila dijalankan dengan program pensinyalan yang dibuat dengan bahasa *assembly*.

1.3 TUJUAN PENELITIAN

Tujuan skripsi ini meliputi : (1) Penguasaan teknologi dasar rancang bangun simulasi sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim berbasis rangkaian mikrokontroler Atmel 89S51. (2) Mampu membuat rancang bangun rangkaian simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim menggunakan mikrokontroler AT89S51. (3) Mampu menganalisis unjuk kerja rangkaian mikrokontroler Atmel 89S51 dalam pemanfaatannya sebagai rangkaian simulasi pada sistem pensinyalan komunikasi seluler.

1.4 BATASAN MASALAH

Batasan masalah dalam skripsi ini adalah sebagai berikut :

1. Pensinyalan komunikasi seluler yang dibahas adalah untuk *Global System for Mobile Communication (GSM)*
2. Bagian pensinyalan yang dibahas dibatasi pada *subscriber* pengirim dengan kondisi *idle* dan panggilan keluar
3. Komponen pensinyalan yang diaplikasikan pada skripsi ini adalah *mobile station (MS)* dan *mobile switching center (MSC)*
4. Program pensinyalan yang dibuat tidak menggunakan prosedur autentikasi
5. Rangkaian simulator yang dibuat hanya digunakan untuk menampilkan bit-bit pensinyalan yang telah teruji validitasnya
6. Komponen pensinyalan MS dan MSC pada rangkaian simulator dibuat tidak terpisah
7. Input nomor ponsel tujuan tidak menggunakan aturan penomoran ponsel yang umum digunakan di Indonesia

1.5 METODOLOGI PENELITIAN

Penelitian dilakukan dengan membuat rancang bangun sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim menggunakan mikrokontroler Atmel 89S51 dengan memenuhi karakteristik *message flow* pensinyalan komunikasi seluler beserta tipe *message* dan bit-bit yang terkandung di dalamnya. Kemudian dilakukan analisis terhadap cara kerja serta performansi dari rangkaian mikrokontroler yang dibuat.

1.6 SISTEMATIKA PENULISAN

Guna membuat pembahasan masalah pada skripsi menjadi lebih sistematis, skripsi ini dibagi menjadi beberapa bab.

Bab pertama, Pendahuluan, memuat latar belakang masalah, tujuan penulisan, pembatasan masalah, metode penelitian, dan sistematika penulisan. Bab kedua, Landasan teori, Menjelaskan dasar teori mengenai sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim, prinsip kerja, elemen dasar dan arsitektur sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim.

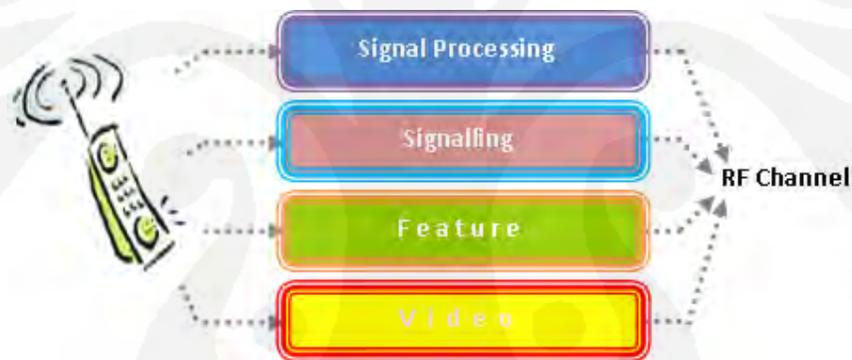
Menjelaskan bahasa pemrograman *assembly*, prinsip kerja dan komponen mikrokontroler Atmel 89S51. Bab ketiga, Rancang Bangun, Menjelaskan algoritma dan program dasar perancangan rangkaian. Menguraikan proses perancangan rangkaian dengan menjelaskan prinsip kerja yang dilakukan tiap sub-sistem penyusun rangkaian utama. Bab keempat, Pengujian dan Analisa, Menganalisis hasil pengujian pada tiap sub-sistem bagian penyusun rangkaian utama mikrokontroler sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim. Menganalisis cara kerja dan unjuk kerja keseluruhan rangkaian mikrokontroler sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim. Bab kelima, Kesimpulan, Bagian ini berisi kesimpulan dari keseluruhan isi skripsi

BAB II

DASAR TEORI

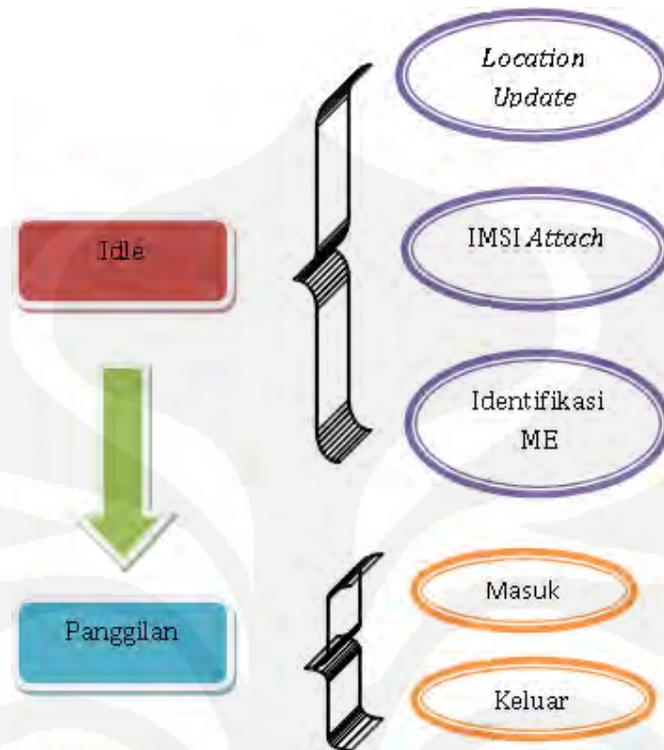
2.1 PENSINYALAN PADA KOMUNIKASI SELULER

Telekomunikasi merupakan proses interaksi antara dua terminal yaitu pengirim dan penerima. Dalam konteksnya, terdapat beberapa fungsi yang mendukung terjadinya hubungan telekomunikasi. Gambar 2.1 menunjukkan fungsi-fungsi pada komunikasi seluler. Fungsi-fungsi tersebut adalah pemrosesan sinyal, pensinyalan, fitur, dan video. Fungsi-fungsi tersebut akan disatukan dalam kanal RF untuk dikirim ke proses selanjutnya.



Gambar 2.1. Fungsi-fungsi pada Komunikasi Seluler.

Perbedaan fungsi *signal processing* dengan *signalling* (pensinyalan) terletak pada input. Untuk *signal processing* inputnya dalam bentuk suara, sedangkan untuk pensinyalan inputnya berupa kode *Dual Tone Multi Frequency* (DTMF). Pada skripsi ini, dilakukan rancang bangun pada proses pensinyalan. Pensinyalan adalah fungsi pengontrolan yang diawali pembentukan koneksi, memulai komunikasi, sampai dengan pengakhiran koneksi. Dalam skripsi ini, akan dibahas secara khusus salah satu fungsi pada komunikasi seluler yaitu pensinyalan. Untuk memahami lebih mudah kondisi-kondisi yang terjadi saat pensinyalan, berikut ini diberikan bagan yang menggambarkan kondisi pensinyalan komunikasi seluler.



Gambar 2.2. Bagan Sederhana Pensinyalan Komunikasi Seluler.

Gambar 2.2 menunjukkan kondisi yang terjadi saat pensinyalan komunikasi seluler bagian *subscriber* pengirim berlangsung. Proses ini diawali dengan kondisi *idle*, dimana *Mobile Station* berada dalam keadaan nyala atau *on* namun tidak melakukan pertukaran informasi, dan diakhiri pada proses pertukaran informasi yang dalam hal ini dibatasi hanya untuk panggilan keluar saja. Berikut ini akan dijelaskan lebih mendalam tentang pensinyalan itu sendiri beserta *message flow* yang terkait.

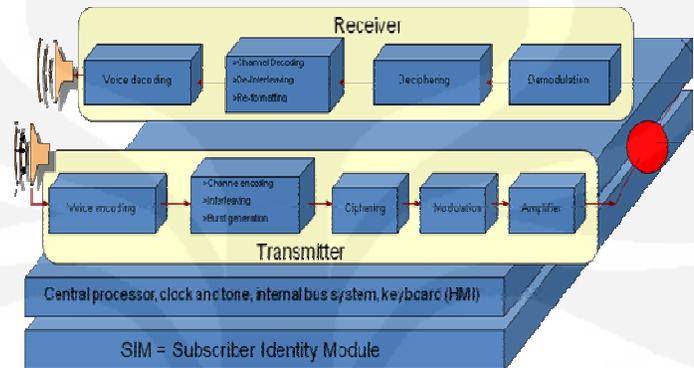
2.1.1 Pensinyalan

Untuk memahami proses pensinyalan, ada beberapa elemen jaringan terkait yang perlu dipahami terlebih dahulu, berikut ini adalah penjelasannya:

- ❖ *Mobile Station (MS)*

MS tersusun atas peralatan bergerak (terminal) atau *Mobile Equipment (ME)* dan suatu *smart card* yang disebut *Subscriber Identity Module (SIM)*. SIM menyediakan mobilitas personal sehingga pengguna dapat mengakses layanan yang diinginkan tanpa tergantung terminalnya. Dengan

menyisipkan kartu SIM ke dalam terminal lain, pengguna dapat menerima panggilan dari terminal tersebut, membuat panggilan dari terminal tersebut dan menerima layanan lainnya. Pada Gambar 2.3 diperlihatkan sebuah blok diagram *mobile station*. Pada gambar tersebut terlihat komponen-komponen dari *mobile station* itu sendiri yaitu *SIM card* dan *mobile equipment*.



Gambar 2.3. Blok Diagram Mobile Station [6]

❖ *Base transceiver Station (BTS)*

Terdiri dari perlengkapan radio yang diperlukan untuk mendukung sebuah sel. Tugas dari BTS adalah menjaga dan memonitor hubungan dengan MS. Lebih khusus lagi, menghubungkan dengan transmisi penerimaan radio, semua fungsi pemrosesan sinyal spesifik dengan radio *interface* dan beberapa fungsi tambahan. BTS juga sering disebut sebagai kepanjangan tangan BSC dan merupakan bagian dengan perangkat keras tersebut. BTS bertugas menjaga dan memonitor hubungan ke *Mobile Station*. Selain itu juga BTS bertugas memancarkan dan menerima data dari/ke MS. BTS mempunyai *transceiver* radio yang mendefinisikan suatu sel dan menangani protokol jalur radio dengan stasiun bergerak. Di daerah urban yang luas, akan banyak terdapat BTS, karena itu BTS harus memenuhi beberapa persyaratan, yaitu tahan banting (*ruggedness*), reliabilitas, portabilitas, dan biaya minimum.

❖ *Mobile Switching Center (MSC)*

MSC melakukan fungsi switching dasar, mengatur BSC melalui A-Interface, sebagai penghubung antara satu jaringan GSM dengan jaringan lainnya melalui *Internetworking Function (IWF)*, dan sebagai gerbang

(*gateway*) ke jaringan lain, dan menghubungkan elemen jaringan NSS dengan elemen jaringan BSS pada suatu *Public Line Mobile Network* (PLMN). MSC terhubung ke MSC lain dalam PLMN yang sama.

❖ *Home Location Register (HLR)*

Berisi rekaman database semi-permanen dan transien dari pelanggan serta merupakan database user yang utama. HLR juga berisi rekaman lengkap lokasi terkini dari user yaitu alamat VLR yang sedang melayani pelanggan yang terdaftar padanya.

❖ *Visitor Location register (VLR)*

Berisi database sementara dari pelanggan dan database dinamis sistem. Data pelanggan di VLR merupakan data transien dan berisi data pelanggan yang sedang berada dalam daerah layanannya. VLR digunakan untuk pelanggan lokal dan yang sedang melakukan roaming. VLR memiliki pertukaran data yang luas daripada HLR. VLR diakses oleh MSC untuk setiap panggilan, dan MSC dihubungkan dengan VLR. Setiap MSC terhubung dengan sebuah VLR, tetapi satu VLR dapat terhubung dengan beberapa MSC.

❖ *Equipment Identity register (EIR)*

Merupakan register penyimpanan data seluruh *Mobile Stations*. EIR berisi IMEIs (*International Mobile Equipment Identities*), yang merupakan nomor seri perangkat dan tipe kode tertentu. EIR digunakan untuk meneliti keabsahan pesawat telepon selular (IME) yang digunakan pelanggan.

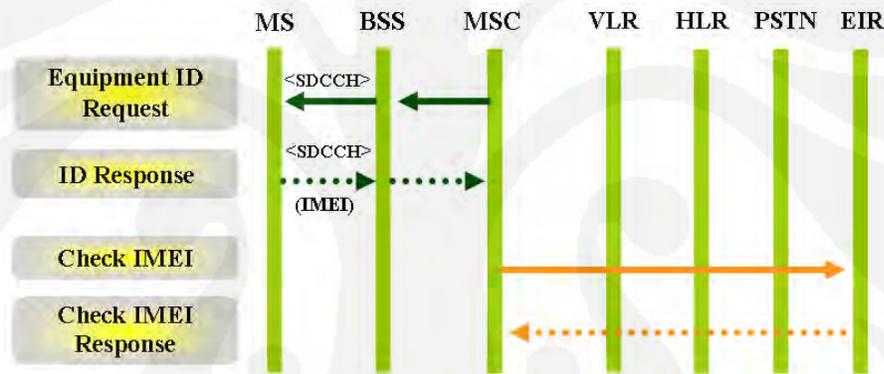
Pensinyalan pada sistem komunikasi seluler terjadi setiap kali MS melakukan koneksi dengan BTS. Pensinyalan dilakukan oleh MS berdasarkan perintah dari pengguna. Saat MS dinyalakan, MS melakukan beberapa proses pendahuluan sebelum dapat berkomunikasi dengan MS lain ataupun dengan jaringan lain, dalam hal ini PSTN (*Public Switched Telephone Network*). Pensinyalan yang akan dibahas dalam skripsi ini dibagi menjadi dua kondisi, yaitu kondisi *idle* dan kondisi panggilan.

2.1.1.1 Kondisi Idle

Setelah MS dapat menentukan keberadaannya pada jangkauan BTS mana, MS akan terus melakukan komunikasi dengan BTS. Keadaan seperti ini adalah keadaan saat *idle*. *Idle* adalah kondisi dimana MS dalam keadaan menyala, dan selalu terjadi pensinyalan dengan BTS. Kondisi *idle* memiliki beberapa perilaku, yaitu identifikasi ME, *cell reselection* dan *location area update*

1. Identifikasi ME

Proses identifikasi ME ini untuk melihat apakah ME yang digunakan berada pada kategori ME yang bisa dilayani atau tidak. Gambar 2.4 memperlihatkan aliran pesan yang terjadi saat identifikasi ME. Pada gambar terlihat bahwa komponen yang terlibat pada proses ini adalah MS, *Base Station Subsystem* (BSS), MSC, VLR, HLR, dan EIR.



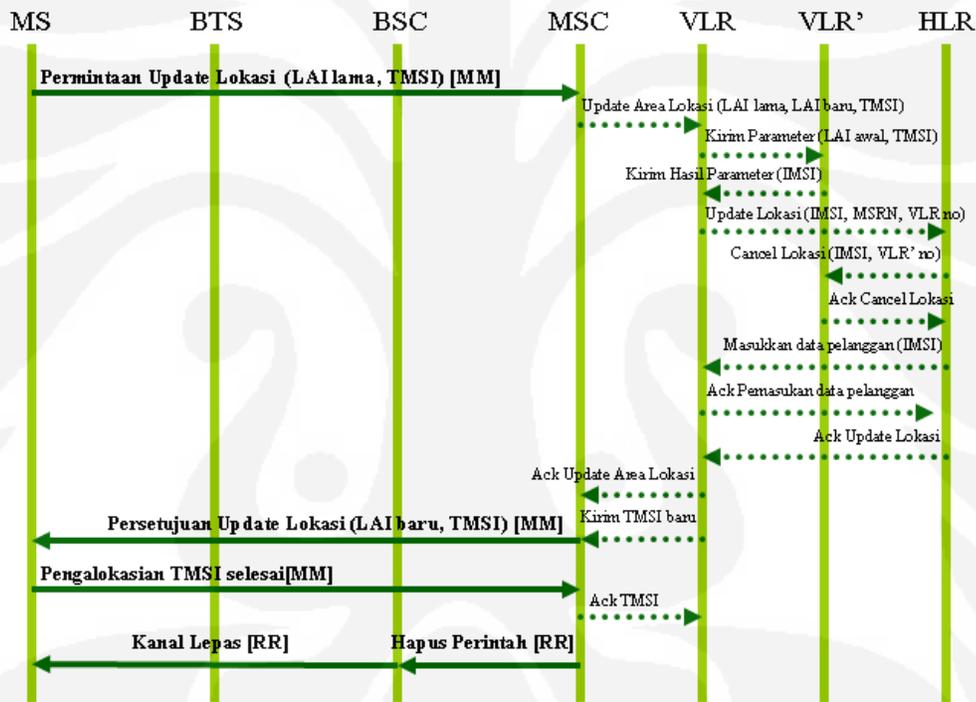
Gambar 2.4. Aliran Pesan saat melakukan Identifikasi ME [5]

2. Cell reselection

Cell reselection pertama dimulai tepat setelah MS dinyalakan, atau biasa disebut dengan IMSI (*International Mobile Subscriber Identity*) *attach*. Proses ini dilakukan untuk menentukan sel yang melayani MS tersebut. Setelah itu MS akan terus melakukan pensinyalan dengan BTS. *Cell reselection* selanjutnya terjadi apabila terdapat sel *neighbor* yang memiliki level sinyal lebih kuat maka MS akan melakukan pemilihan ulang sel, atau biasa disebut *cell reselection*. Pada prosedur ini MS selalu memantau semua BCCH *carrier* sel *neighbor* untuk mengetahui jika ada sel lain yang mungkin ditempati.

3. Location Update

MS tidak selalu berada pada satu area jangkauan suatu BTS, oleh karena itu ada prosedur yang dinamakan *location area update*. Perbaruan lokasi ini dilakukan dengan melihat LAI pada MS, jika sudah tidak sama, maka MS akan meng-*update* lokasi. Perbaruan lokasi ini, dimulai dengan permintaan *location update* dari MS pada VLR yang baru. Lalu MSC akan mengirimkan nomor TMSI sebagai identitas sementara pengganti IMSI, yang disimpan pada MS maupun VLR. Gambar 2.5 memperlihatkan aliran pesan yang terjadi saat proses perbaruan lokasi. Pada proses ini komponen yang terlibat adalah MS, BSS, MSC, VLR, dan HLR.



Gambar 2.5. Aliran Pesan saat Perbaruan Lokasi [5]

Tahap-tahap yang harus dilewati untuk melakukan perbaruan lokasi dijelaskan sebagai berikut :

1. MS mengirim pesan permintaan kanal ke BTS dan diteruskan ke BSC.,
2. BSC menyediakan kanal pensinyalan yaitu SDCCH, jika ada satu yang tidak terpakai, dan memberitahu BTS untuk mengaktifkannya.,
3. MS diberitahu untuk men-*set* SDCCH,

4. MS mengirim pesan permintaan *update* lokasi yang terdiri dari identitas MS, identitas area lokasi yang lama dan SABM.,
5. MS juga mengirim parameter autentikasi, jika MS sudah terdaftar pada MSC (dalam hal ini VLR) dan parameter autentikasi yang digunakan disimpan di VLR. Sedangkan untuk MS yang belum terdaftar pada VLR, maka HLR atau VLR yang digunakan sebelumnya harus menghubungi untuk meminta data pelanggan MS dan parameter autentikasi.,
6. MS mengirim jawaban menggunakan parameter autentikasi yang diterima. Jika sukses, VLR akan diperbarui.,
7. MS menerima persetujuan *update* lokasi,
8. MS diberitahu untuk melepaskan SDCCH dan berubah ke mode *idle*.

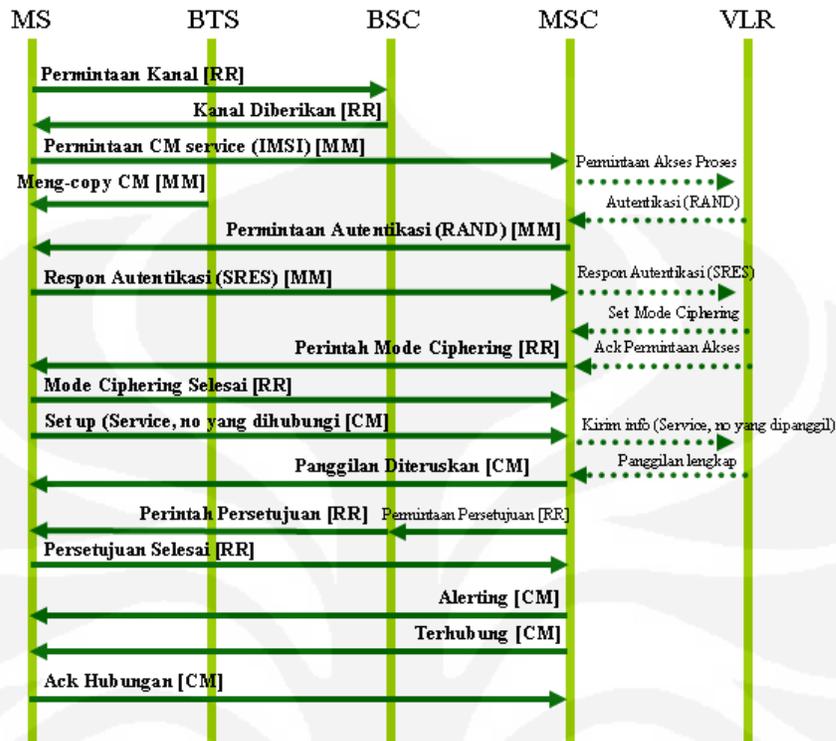
2.1.1.2 Panggilan

Panggilan adalah kondisi dimana MS akan melakukan pertukaran informasi, baik dengan MS lain maupun dengan PSTN. Sebelum melakukan panggilan terlebih dahulu dilakukan *set up* panggilan. Proses setup panggilan dimulai dengan pembuatan hubungan antara MSC dan BSC. Dan diakhiri tepat saat pemanggil memutuskan panggilan.

2.1.1.2.1 Panggilan Keluar

Sekali MS sukses terhubung ke jaringan GSM, panggilan dapat dilakukan dari MS ke MS lain pada PSTN. Pelanggan hanya menekan nomor telepon, lalu tekan tombol *dial*, dan MS akan mengirim pesan permintaan set up panggilan ke jaringan PLMN melalui BTS. Elemen yang menangani permintaan panggilan adalah MSC. MSC memeriksa VLR apakah panggilan keluar dibolehkan. Jika ya, MSC akan mencari tujuan panggilan.

Perbedaan saat melakukan panggilan ke PSTN terletak pada beralihnya indikasi panggilan melalui GMSC yang biasanya berada pada MSC untuk keluar dari jaringan PLMN ke jaringan PSTN. Gambar 2.6 memperlihatkan aliran pesan yang terjadi saat proses panggilan keluar. Komponen yang terlibat pada proses ini adalah MS, BSS, dan MSC.



Gambar 2.6. Aliran Pesan Panggilan Keluar [5]

Kanal yang digunakan untuk meminta kanal adalah RACH. RACH dapat digunakan oleh semua MS yang ada, tanpa koordinasi terlebih dahulu, sehingga sangat memungkinkan terjadinya tabrakan. Setelah itu MS melihat pada AGCH untuk mengetahui balasannya. AGCH berisi nomor SDDCH pada BTS, *frame unit* berapa dan TS berapa. Setelah itu dilakukan proses autentikasi. Apabila autentikasi sukses, maka MSC mengirimkan perintah *ciphering* (penyandian). Hubungan diakhiri apabila pelanggan yang melakukan panggilan sudah memutuskan koneksi.

2.1.1.2.2 Panggilan Masuk

◆ Tahap Pertama (Menghubungi GMSC)

Ketika seseorang dari PSTN melakukan panggilan ke MS, mereka menekan nomor telepon (MSISDN) dan panggilan akan di-*routing* ke operator GMSC. GMSC berfungsi sebagai jalan masuk dari bagian luar PLMN (PSTN atau PLMN lain) ke jaringan PLMN tersebut.

GMSC adalah untuk menentukan lokasi saat ini dari MS dalam rangka untuk menyambungkan panggilan. GMSC melakukan ini dengan berkonsultasi dengan HLR.

◆ Tahap Kedua (*Routing Panggilan*)

Ketika HLR menerima pesan meragukan, maka yang dilakukan adalah panggilan harus di-*routing* ke nomor lain (dialihkan), atau jika secara langsung di-*routing* ke MS.

Jika pemilik MS telah meminta untuk mengalihkan semua panggilan ke nomor lain, disebut sebagai nomor CFU (*Call Forwarding Unconditional*), maka panggilan akan dikembalikan ke GMSC untuk *routing* ke nomor CFU.

Jika MS tidak secara langsung terhubung dengan VLR (karena MS dimatikan), maka HLR akan memberikan nomor yang diketahui sebagai nomor CFNRc (*Call Forwarding Number Reachable*) ke GMSC sehingga panggilan akan diteruskan ke nomor CFNRc.

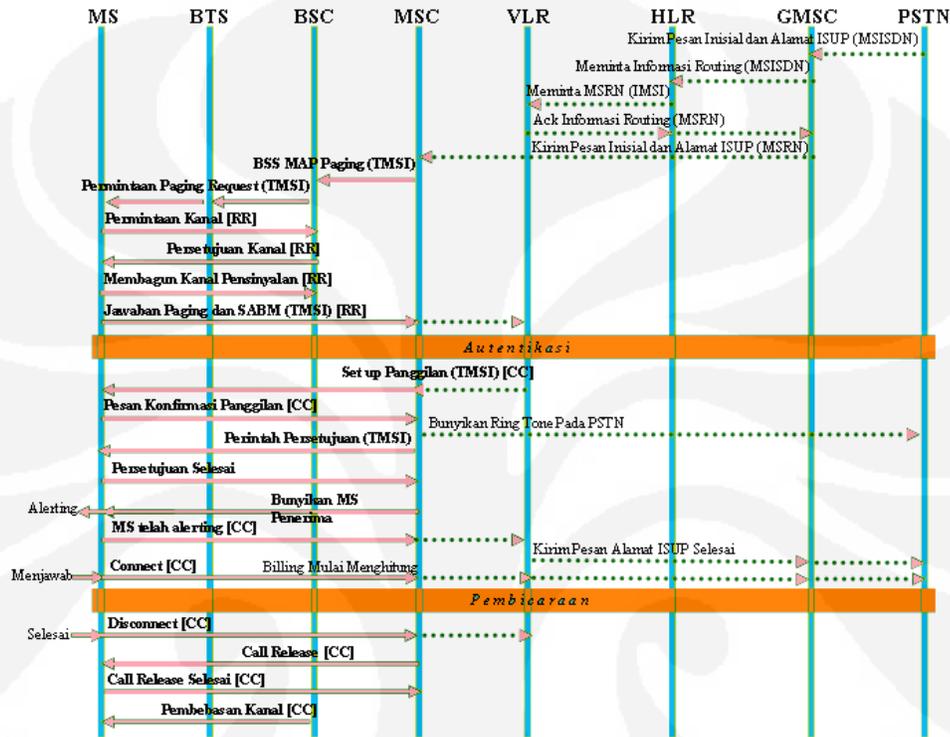
Akhirnya, jika HLR mengetahui bahwa MS tersebut berada dalam yurisdiksi terutama dari VLR, maka MS akan meminta nomor sementara (MSRN) dari VLR tersebut. Nomor ini disampaikan ke GMSC, yang digunakan untuk *routing* panggilan ke MSC lain, dinamakan MSC yang dikunjungi.

◆ Tahap Ketiga (Membunyikan Telepon)

Ketika panggilan diterima oleh MSC yang dikunjungi, MSRN digunakan untuk menemukan rekaman MS pada VLR. Rekaman ini menunjukkan area lokasi MS tujuan. *Paging* terjadi pada semua BTS di area tersebut. Ketika MS pelanggan merespon, lokasi dari MS tujuan dikembalikan ke MSC yang dikunjungi. VMSC lalu meneruskan panggilan ke BTS yang semestinya, dan MS tujuan akan berbunyi. Jika pelanggan menjawab, kanal suara akan terbentuk melewati VMSC dan GMSC kembali ke jaringan area pelanggan yang membuat panggilan.

Jika MS tujuan tidak menjawab, karena pelanggan sedang sibuk dengan nomor lain (*call waiting* tidak digunakan), VMSC *routing*

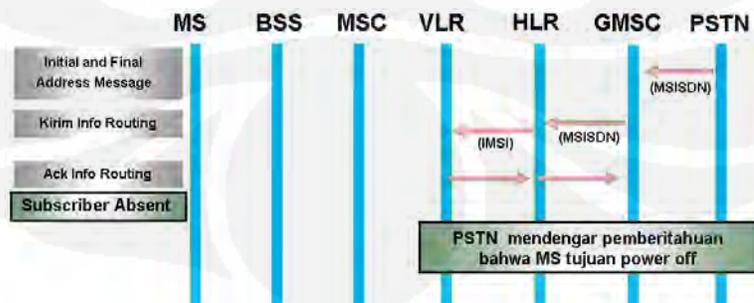
panggilan ke nomor CFB (*Call Forward Busy*). Demikian pula, jika pelanggan tidak menjawab panggilan setelah beberapa periode waktu (biasanya 30 detik) lalu VMSC *routing* panggilan ke nomor CFNRy (*Call Forwarding No Reply*).



Gambar 2.7. Aliran Pesan Panggilan Masuk

Gambar 2.7 memperlihatkan aliran pesan yang terjadi saat proses panggilan masuk. *Message Flow* diatas menandakan bahwa MS penerima dapat melakukan hubungan. Bagaimana jika tidak? Berikut ini ditampilkan beberapa *Message Flow* apabila MS tidak dapat dihubungi.

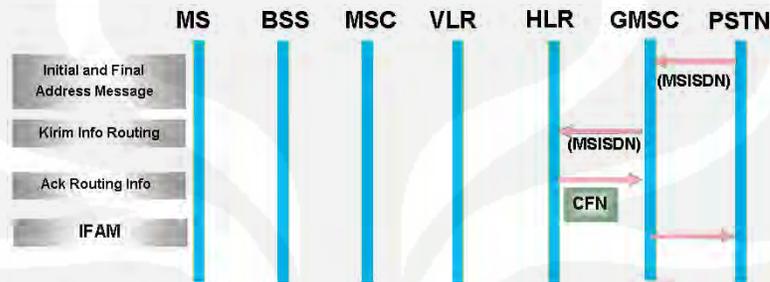
◆ MS dalam keadaan *off*



Gambar 2.8. Message Flow saat MS tujuan Mati [6]

Gambar 2.8 diatas memperlihatkan aliran pesan yang terjadi saat MS tujuan mati atau dalam keadaan *off*. Komponen yang terlibat saat proses ini adalah VLR, HLR, GMSC, dan PSTN. Saat PSTN memanggil MS tujuan namun MS tujuan dalam keadaan *off*, maka PSTN akan menerima pesan bahwa MS tujuan sedang tidak aktif.

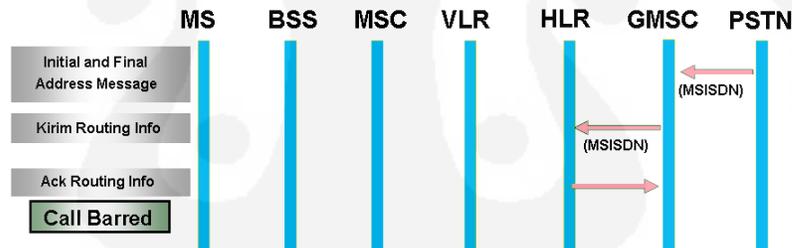
◆ MS menggunakan nomor CFU



Gambar 2.9. Message Flow saat MS mengaktifkan CFN [6]

Gambar 2.9 memperlihatkan saat MS mengaktifkan nomor CFU kepada nomor PSTN yang memanggil. Komponen yang terlibat adalah HLR, GMSC, dan PSTN.

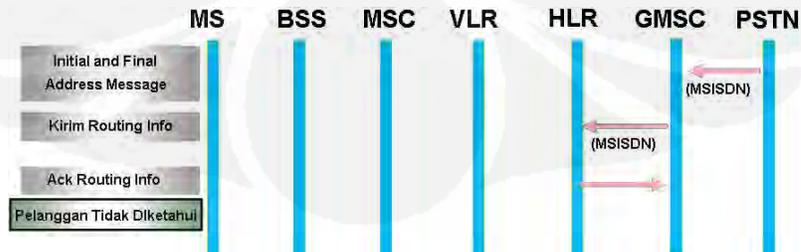
◆ MS menolak semua panggilan masuk



Gambar 2.10. Message Flow saat MS Menolak Semua Panggilan Masuk [6]

Gambar 2.10 memperlihatkan aliran pesan yang terjadi saat MS menolak semua panggilan masuk. Komponen yang terlibat pada proses ini adalah HLR, GMSC, dan PSTN.

◆ MS tidak diketahui di HLR

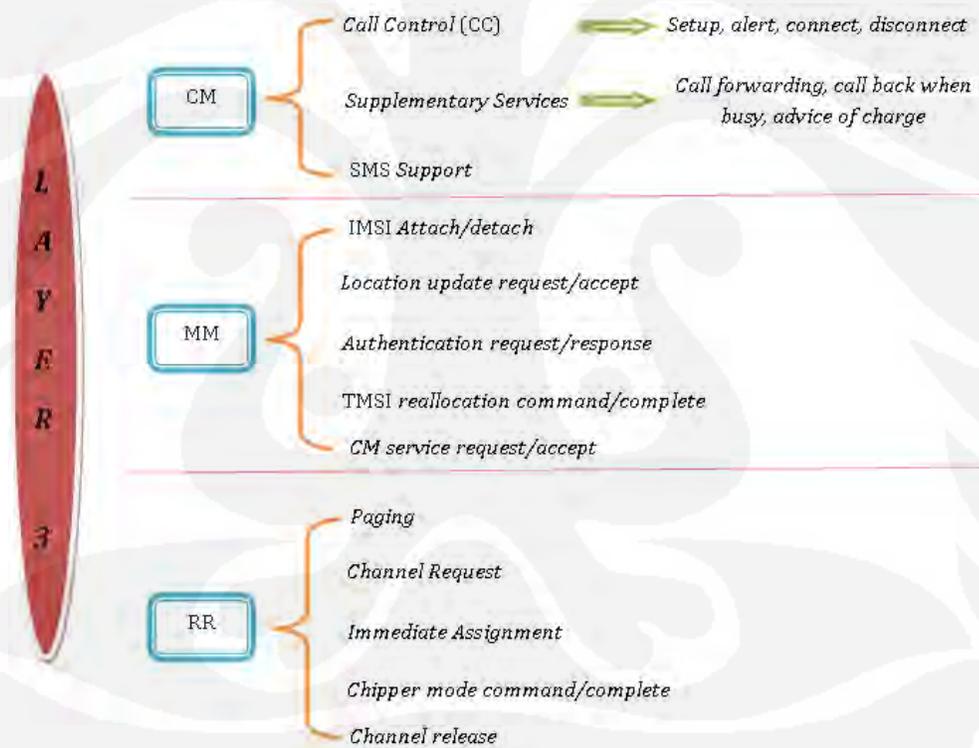


Gambar 2.11. Message Flow saat MS tidak diketahui di HLR [6]

Gambar 2.11 memperlihatkan aliran pesan yang terjadi saat MS tidak diketahui di HLR. Komponen yang terlibat adalah HLR, GMSC, dan PSTN.

2.1.2 Fungsi-fungsi Layer 3 Pensinyalan

Protokol pada jaringan komunikasi seluler yang berkaitan dengan pensinyalan adalah *layer 3* yaitu *network layer* yang memiliki fungsi-fungsi diantaranya *radio resource management (RR)*, *mobility management (MM)*, dan *connection management (CM)*. Fungsi-fungsi ini yang nantinya akan berperan besar selama proses pensinyalan berlangsung pada jaringan komunikasi seluler. Untuk memahami lebih jelas mengenai fungsi-fungsi tersebut, berikut ini adalah bagan sederhana yang menggambarkan kedudukan fungsi tersebut dalam proses pensinyalan komunikasi seluler.



Gambar 2.12. Fungsi-fungsi protokol layer 3 pada pensinyalan komunikasi seluler

Gambar 2.12 diatas memperlihatkan bagan sederhana kedudukan fungsi-fungsi protokol *layer 3* pada proses pensinyalan komunikasi seluler. Penjelasan masing-masing fungsi tersebut akan diberikan berikut ini.

2.1.2.1 Connection Management (CM)

Fungsi CM dapat digunakan untuk kontrol panggilan (*Call Control*), pengelolaan layanan tambahan (*Supplementary Services*), dan pengelolaan layanan pesan singkat (*SMS Support*). *Call control* (CC) bertanggung jawab pada proses membuat, menjaga, dan mengakhiri panggilan. Selain itu CC juga menangani kondisi *setup*, *alert*, *connect*, dan *disconnect*. Kegunaan lain dari CM adalah pengelolaan layanan tambahan. *Supplementary Services* (SS) bertanggung jawab untuk layanan meneruskan panggilan (*call forwarding*), memanggil kembali saat kondisi sibuk, dan lain-lain. Salah satu fungsi CM yang paling penting adalah *call routing* dalam rangka untuk mencari MS pelanggan.

2.1.2.2 Mobility Management (MM)

Fungsi MM berkaitan dengan semua aspek yang terkait dengan mobilitas pengguna, terutama pengelolaan lokasi serta autentikasi dan keamanannya. Ketika MS dinyalakan, MS melakukan prosedur *update* lokasi dengan menunjukkan IMSI ke jaringan. Selain itu MS juga menunjukkan lokasi terbaru MS tersebut.

Prosedur autentikasi melibatkan kartu SIM dan pusat autentikasi (AuC). Pada MS dan jaringan sama-sama menyimpan kode rahasia (Ki). Dengan menggunakan A3, dari Ki dan RAND akan menghasilkan SRES (*Signed Response*). Perhitungan SRES di SIM dan AuC inilah yang akan dibandingkan.

2.1.2.3 Radio Resource Management (RR)

Fungsi RR untuk membangun, menjaga, dan mengakhiri komunikasi antara MS dengan MSC. Elemen jaringan yang terkait dengan fungsi RR adalah MS dan BTS. RR juga bertanggung jawab untuk pengaturan spektrum frekuensi dan reaksi jaringan terhadap pergantian kondisi lingkungan radio. Beberapa prosedur utama dari RR diantaranya penentuan, pergantian, dan pengakhiran kanal, *handover*, kontrol level daya, dan *frequency hopping*.

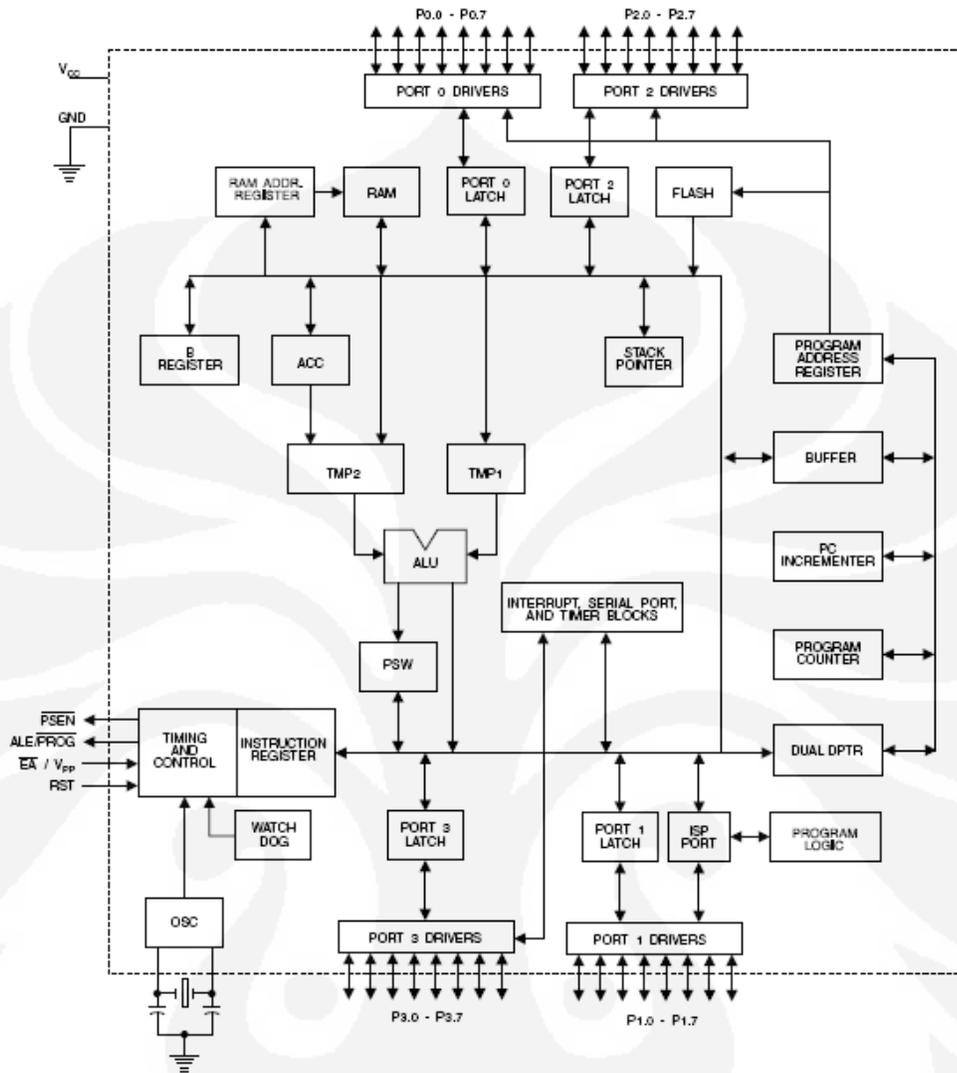
2.2 MIKROKONTROLLER ATMEL 89S51 (AT89S51)

2.2.1 Deskripsi Sistem

AT89S51 membutuhkan konsumsi daya rendah, mikrokontroler 8-bit CMOS dengan 4K byte memori *Flash ISP* (*in system programmable*/ dapat diprogram di dalam sistem). Divais ini dibuat dengan teknologi memori *nonvolatile* kerapatan tinggi dan kompatibel dengan standar industri 8051, set instruksi dan pin keluaran. *Flash* yang berada di dalam *chip* memungkinkan memori program untuk diprogram ulang pada saat *chip* di dalam sistem atau dengan menggunakan *Programmer* memori *nonvolatile* konvensional. Dengan mengkombinasikan CPU 8 bit yang serba guna dengan *flash ISP* pada *chip*, ATMEL 89S51 merupakan mikrokontroler yang memberikan fleksibilitas yang tinggi dan penyelesaian biaya yang efektif untuk beberapa aplikasi kontrol.

AT89S51 mempunyai fitur-fitur standar sebagai berikut: 4K byte *Flash*, 128 byte RAM, 32 jalur I/O, *Timer Watchdog*, dua data *pointer*, dua 16 bit *timer/ counter*, lima vektor interupsi dua level, sebuah *port serial full* dupleks, osilator internal, dan rangkaian *clock*. Selain itu AT89S51 didisain dengan logika statis untuk operasi dengan frekuensi sampai 0 Hz dan didukung dengan mode penghematan daya. Pada saat keadaan *idle*, sistem akan menghentikan CPU, sementara RAM, *timer/ counter*, serial *port* dan sistem interupsi tetap berfungsi. Mode *Power Down* akan tetap menyimpan isi dari RAM tetapi akan membekukan osilator, menggagalkan semua fungsi chip sampai interupsi eksternal atau *reset hardware* terjadi.

Gambar 2.13 menunjukkan blok diagram dari AT89S51.



Gambar 2.13. Blok Diagram AT89S51 [11]

2.2.2 Deskripsi Pin

Berikut ini adalah penjelasan pin-pin yang terdapat pada mikrokontroler AT89S51 :

- a. **VCC** : Tegangan Supply
- b. **GND** : Ground
- c. **Port 0 (P0)**

Port 0, merupakan *port I/O* 8 bit *open drain* dua arah. Sebagai sebuah *port*, setiap pin dapat mengendalikan 8 input TTL. Ketika logika “1” dituliskan ke port 0, maka port dapat digunakan sebagai input dengan *high*

impedansi. Port 0 dapat juga dikonfigurasi untuk *multiplexing* dengan *address/data bus* selama mengakses memori program atau data eksternal. Pada mode ini P0 harus mempunyai *pull up*.

d. Port 1 (P1)

Port 1 merupakan *port I/O* 8 bit dua arah dengan internal *pull up*. *Buffer output port 1* dapat mengendalikan empat TTL input. Ketika logika “1” dituliskan ke *port 1*, maka *port* ini akan bersifat internal *pull up* dan dapat digunakan sebagai input. *Port 1* juga menerima alamat *byte* rendah selama pemrograman dan verifikasi *Flash*. Dalam *port 1* juga terdapat *port pin* fungsi alternative, yaitu :

- ❖ P1.5 MOSI (digunakan untuk *In System Programming*).
- ❖ P1.6 MISO (digunakan untuk *In System Programming*).
- ❖ P1.7 SCK (digunakan untuk *In System Programming*).

e. Port 2

Port 2 merupakan *port I/O* 8 bit dua arah dengan internal *pull up*. *Buffer output port 2* dapat mengendalikan empat TTL input. Ketika logika “1” dituliskan ke *port 2*, maka *port* ini akan mendapatkan internal *pull up* dan dapat digunakan sebagai input.

f. Port 3

Port 3 merupakan *port I/O* 8 bit dua arah dengan internal *pull up*. *Buffer output port 3* dapat mengendalikan empat TTL input. Ketika logika “1” dituliskan ke *port 3*, maka *port* ini akan mendapatkan internal *pull up* dan dapat digunakan sebagai input. *Port 3* juga melayani berbagai macam fitur khusus, sebagaimana yang ditunjukkan pada tabel 2.1 berikut:

Tabel 2.1 Fitur-fitur khusus pada *port 3* [11]

Port Pin	Fungsi Alternatif
P3.0	RXD (<i>port serial input</i>)
P3.1	TXD (<i>port serial output</i>)
P3.2	INT0 (interupsi eksternal 0)
P3.3	INT1 (interupsi eksternal 1)
P3.4	T0 (input eksternal <i>timer 0</i>)
P3.5	T1 (input eksternal <i>timer 1</i>)
P3.6	WR (<i>write strobe</i> memori data eksternal)
P3.7	WR (<i>read strobe</i> memori program eksternal)

g. RST

Input *Reset*. Logika *high* “1” pada pin ini untuk dua siklus mesin sementara osilator bekerja maka akan me-*reset* divais.

h. ALE/PROG

Address Latch Enable (ALE) merupakan suatu pulsa output untuk mengunci *byte low* dari alamat selama mengakses memori eksternal. Pin ini juga merupakan input pulsa pemrograman selama pemrograman *flash* (paralel). Pada operasi normal, ALE mengeluarkan suatu laju konstan 1/6 dari frekuensi osilator dan dapat digunakan untuk pewaktu eksternal.

i. PSEN

Program Store Enable merupakan *strobe read* untuk memori program eksternal.

j. EA/VPP

Merupakan singkatan dari *Eksternal Access Enable*. EA harus di hubungkan ke GND untuk *enable* divais, untuk memasuki memori program eksternal mulai alamat 0000H s/d FFFFH. EA harus dihubungkan ke VCC untuk akses memori program internal Pin ini juga menerima tegangan pemrograman (VPP) selama pemrograman *Flash*.

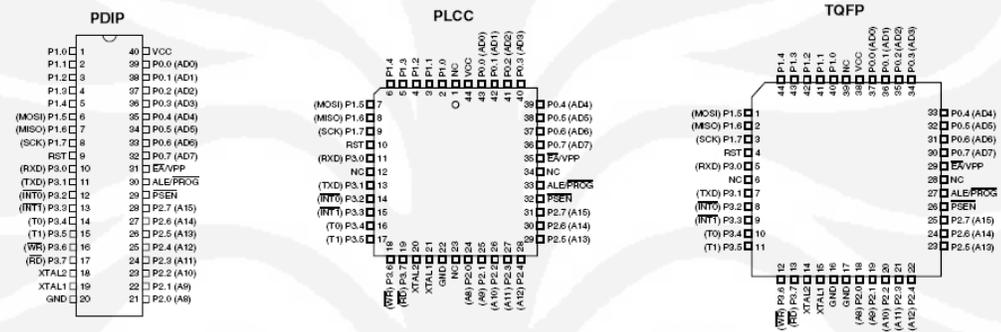
k. XTAL1

Input untuk penguat osilator *inverting* dan input untuk rangkaian internal *clock*.

l. XTAL2

Output dari penguat osilator *inverting*.

Gambar 2.14 menunjukkan konfigurasi pin yang terdapat pada mikrokontroler AT89S51.



Gambar 2.14. Konfigurasi Pin AT89S51 [11]

Dalam membuat program-program pensinyalan yang akan digunakan pada mikrokontroler AT89S51, digunakan bahasa assembly sebagai bahasa pemrograman. Berikut ini akan dijelaskan mengenai bahasa assembly yang digunakan.

2.3 BAHASA PEMROGRAMAN ASSEMBLY UNTUK AT89S51

Secara fisik, kerja dari sebuah mikrokontroler dapat dijelaskan sebagai siklus pembacaan instruksi yang tersimpan di dalam memori. Mikrokontroler menentukan alamat dari memori program yang akan dibaca, dan melakukan proses pembacaan data di memori. Data yang dibaca diinterpretasikan sebagai instruksi. Alamat instruksi disimpan oleh mikrokontroler di *register*, yang dikenal sebagai program *counter*. Instruksi ini misalnya program aritmatika yang melibatkan 2 *register*. Sarana yang ada dalam program *assembly* sangat minim, tidak seperti dalam bahasa pemrograman tingkat atas (*high level language*

programming) yang semuanya sudah siap pakai. Penulis program *assembly* harus menentukan segalanya, menentukan letak program yang ditulisnya dalam memori program, membuat data konstan dan tabell konstan dalam memori program, membuat *variabel* yang dipakai kerja dalam memori data dan lain sebagainya.

2.3.1 Program Sumber Assembly

Program sumber *assembly* (*assembly source program*) merupakan kumpulan dari baris-baris perintah yang ditulis dengan program penyunting teks (*text editor*) sederhana, misalnya program EDIT.COM dalam DOS, atau program NOTEPAD dalam Windows atau MIDE-51. Kumpulan baris perintah tersebut biasanya disimpan ke dalam *file* dengan nama ekstensi *.ASM dan lain sebagainya, tergantung pada program *Assembler* yang akan dipakai untuk mengolah program sumber *assembly* tersebut. Setiap baris perintah merupakan sebuah perintah yang utuh, artinya sebuah perintah tidak mungkin dipecah menjadi lebih dari satu baris. Satu baris perintah bisa terdiri atas 4 bagian, bagian pertama dikenali sebagai label atau sering juga disebut sebagai simbol, bagian kedua dikenali sebagai kode operasi, bagian ketiga adalah *operand* dan bagian terakhir adalah komentar. Antara bagian-bagian tersebut dipisahkan dengan sebuah spasi atau tabulator. Berikut ini akan dijelaskan mengenai bagian-bagian dari baris perintah yang terdapat pada program sumber *assembly*.

2.3.1.1 Bagian Label

Label dipakai untuk memberi nama pada sebuah baris perintah, agar bisa mudah menyebutnya dalam penulisan program. Label bisa ditulis apa saja asalkan diawali dengan huruf, biasanya panjangnya tidak lebih dari 16 huruf.

2.3.1.2 Bagian Kode Operasi

Kode operasi (*operation code* atau sering disingkat sebagai *OpCode*) merupakan bagian perintah yang harus dikerjakan. Dalam hal ini dikenal dua macam kode operasi, yang pertama adalah kode operasi untuk mengatur kerja mikroprosesor/mikrokontroler. yang kedua dipakai untuk mengatur kerja program *assembler*, sering dinamakan sebagai *assembler directive*. Kode operasi ditulis

dalam bentuk *mnemonic*, yakni bentuk singkatan singkatan yang relatif mudah diingat, misalnya adalah MOV, ACALL, RET dan lain sebagainya. Kode operasi ini ditentukan oleh pabrik pembuat mikroprosesor/mikrokontroler, dengan demikian setiap prosesor mempunyai kode operasi yang berlainan. Kode operasi berbentuk *mnemonic* tidak dikenal mikroprosesor/mikrokontroler, agar program yang ditulis dengan kode *mnemonic* bisa dipakai untuk mengendalikan prosesor, program semacam itu diterjemahkan menjadi program yang dibentuk dari kode operasi kode biner, yang dikenali oleh mikroprosesor/mikrokontroler. Tugas penerjemahan tersebut dilakukan oleh program yang dinamakan sebagai Program *Assembler*.

2.3.1.3 Bagian Operand

Operand merupakan pelengkap bagian kode operasi, namun tidak semua kode operasi memerlukan *operand*, dengan demikian bisa terjadi sebuah baris perintah hanya terdiri dari kode operasi tanpa *operand*. Sebaliknya ada pula kode operasi yang perlu lebih dari satu *operand*, dalam hal ini antara *operand* satu dengan yang lain dipisahkan dengan tanda koma.

2.3.1.4 Bagian Komentar

Bagian komentar merupakan catatan-catatan penulis program, bagian ini meskipun tidak mutlak diperlukan tapi sangat membantu masalah dokumentasi. Membaca komentar-komentar pada setiap baris perintah, dengan mudah bisa mengerti maksud dan tujuan baris bersangkutan, hal ini sangat membantu orang lain yang membaca program. Bentuk program *assembly* yang umum tampak pada tabel 2.2 sebagai berikut :

Tabel 2.2 Bentuk program *assembly* umum [11]

Label/Simbol	Opcode	Operand	Komentar
	Org	0H	

Start:	Mov	A, #11111110b	; Isi
	Mov	R0, #7	Akumulator
Kiri:	Mov	P0, A	; Isi R0 dengan
	Call	Delay	7
	RL	A	; Copy A ke P0
	DEC	R0	; Panggil Delay
	CJNE	R0, #0, Kiri	
	Sjmp	Start	
Delay:	mov	R1, #255	
Del1:	mov	R2, #255	
Del2:	djnz	R2, del2	
	djnz	R1, del1	
	ret		
	end		

Isi memori adalah bilangan heksadesimal yang dikenal oleh mikrokontroler, yang merupakan representasi dari bahasa *assembly* yang telah kita buat. *Mnemonic* atau *opcode* adalah kode yang akan melakukan aksi terhadap *operand*. *Operand* adalah data yang diproses oleh *opcode*. Sebuah *opcode* bisa membutuhkan satu, dua atau lebih *operand*, bahkan terkadang juga tidak perlu *operand*. Sedangkan komentar dapat kita berikan dengan menggunakan tanda titik koma (;). Berikut contoh jumlah *operand* yang berbeda beda dalam suatu *assembly*.

CJNE R5,#22H, aksi ;dibutuhkan 3 buah operand.

MOVX @DPTR, A ;dibutuhkan 2 buah operand.

RL A ;1 buah operand.

NOP ; tidak memerlukan operand

BAB III

RANCANG BANGUN RANGKAIAN SIMULATOR PENSINYALAN KOMUNIKASI SELULER DENGAN MIKROKONTROLLER AT89S51

3.1 KONSEP RANCANG BANGUN RANGKAIAN SIMULATOR PENSINYALAN KOMUNIKASI SELULER

Simulator pensinyalan komunikasi seluler dibuat dengan menggunakan mikrokontroller AT89S51. Sebelum membuat rangkaian simulator dengan mikrokontroller AT89S51, terlebih dahulu dibuat program-program untuk masing-masing komponen pensinyalan, dalam hal ini yang dilibatkan adalah *Mobile Station* dan *Mobile Switching Center*. Program-program pensinyalan dibuat berdasarkan *message flow* pensinyalan yang secara umum digunakan di Indonesia. Dari *message flow* yang didapatkan berdasarkan *literature* (studi pustaka) dan data dari salah satu *vendor* telekomunikasi di Indonesia, dibuat *flow chart* dan algoritma pensinyalan yang nantinya akan dijadikan dasar dalam pembuatan program-program pensinyalan. Dalam menginterpretasikan *message flow* ke dalam algoritma dan program-program pensinyalan, bit-bit dan *message type* yang dibuat disesuaikan dengan bit-bit dan *message type* yang telah diimplementasikan di Indonesia sehingga diharapkan simulator yang ada benar-benar merepresentasikan konsep pensinyalan komunikasi seluler yang ada di Indonesia. Bit-bit dan *message type* yang merepresentasikan pensinyalan komunikasi seluler dapat dilihat pada Lampiran 1.

Dalam pembuatan program-program pensinyalan, digunakan perangkat lunak simulator 8051IDE. Program-program pensinyalan ini dibuat dengan bahasa pemrograman *assembly* yang cocok diterapkan dengan mikrokontroller AT89S51. Seperti telah dijelaskan pada bagian dasar teori, program sumber *assembly* (*source code*) salah satu bagiannya terdiri dari kode operasi, yaitu kode *mnemonic* yang tidak dapat dikenali oleh mikrokontroller, sehingga untuk menerjemahkan kode *mnemonic* menjadi kode biner yang dikenali mikrokontroller, digunakan program

penerjemah atau program *assembler* yang dalam hal ini adalah simulator 8051IDE.

Tahapan-tahapan yang dilakukan dalam rancang bangun rangkaian simulator pensinyalan ini adalah sebagai berikut :

1. Membuat *flow chart* dan algoritma pensinyalan berdasarkan *message flow* yang didapatkan dari studi pustaka dan data dari salah satu *vendor* telekomunikasi di Indonesia.
2. Membuat program pensinyalan dalam bahasa *assembly* berdasarkan algoritma pensinyalan.
3. Menerjemahkan kode *mnemonic* program pensinyalan menjadi kode biner yang dapat dikenali mikrokontroler dengan menggunakan perangkat lunak 8051IDE.
4. Membuat rangkaian simulator pensinyalan menggunakan mikrokontroler dan komponen-komponen elektronika.
5. Memasukkan program yang telah diterjemahkan ke dalam mikrokontroler AT89S51. Proses ini dikenal dengan nama *burning*.
6. Menjalankan program yang telah dimasukkan ke mikrokontroler. proses *running* ini dilakukan pada mikrokontroler.
7. Melihat dan menganalisa keluaran yang muncul pada rangkaian mikrokontroler.

Untuk mengetahui lebih jelas mengenai tahapan-tahapan yang dilakukan dalam rancang bangun rangkaian simulator pensinyalan komunikasi seluler menggunakan mikrokontroler AT89S51, berikut ini adalah penjelasannya secara detail.

3.1.1 Program-program Pensinyalan

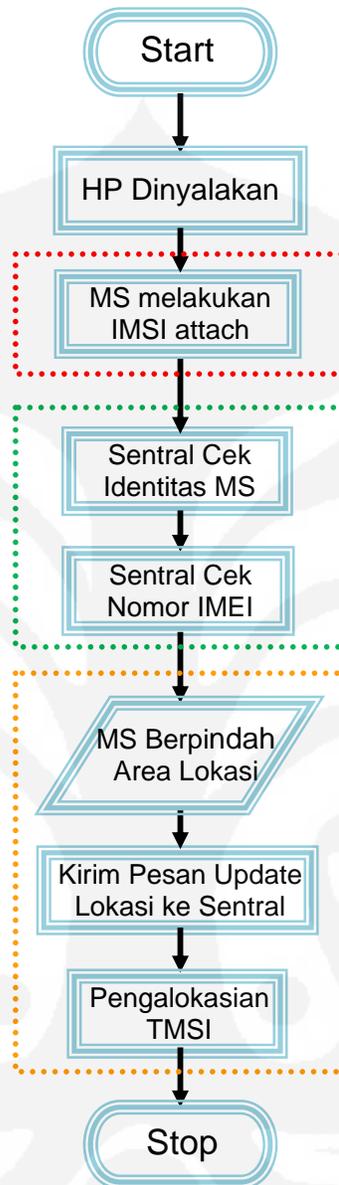
Sebelum membahas lebih jauh mengenai program-program yang dibuat untuk simulator pensinyalan, terlebih dahulu akan dijelaskan mengenai *flow chart* dan algoritma pensinyalan.

Dalam skripsi ini, pensinyalan yang dibahas adalah yang terjadi antara sisi *Mobile Station* (MS) dengan *Mobile Switching Center* (MSC). Bagian ini secara umum dikenal dengan bagian *subscriber* pengirim. Kondisi yang terjadi pada saat

pensinyalan terjadi pada bagian ini adalah kondisi *idle* (*Mobile Station* tidak melakukan aktivitas apapun namun berada dalam keadaan nyala atau *on*) dan kondisi saat MS melakukan pengiriman informasi, dalam hal ini dibatasi pada saat MS melakukan panggilan keluar. Sehingga dari kedua kondisi tersebut dapat dibuat *flow chart* dan algoritma pensinyalan yang menjadi dasar pembuatan program-program pensinyalan.

3.1.1.1 Flow Chart Pensinyalan

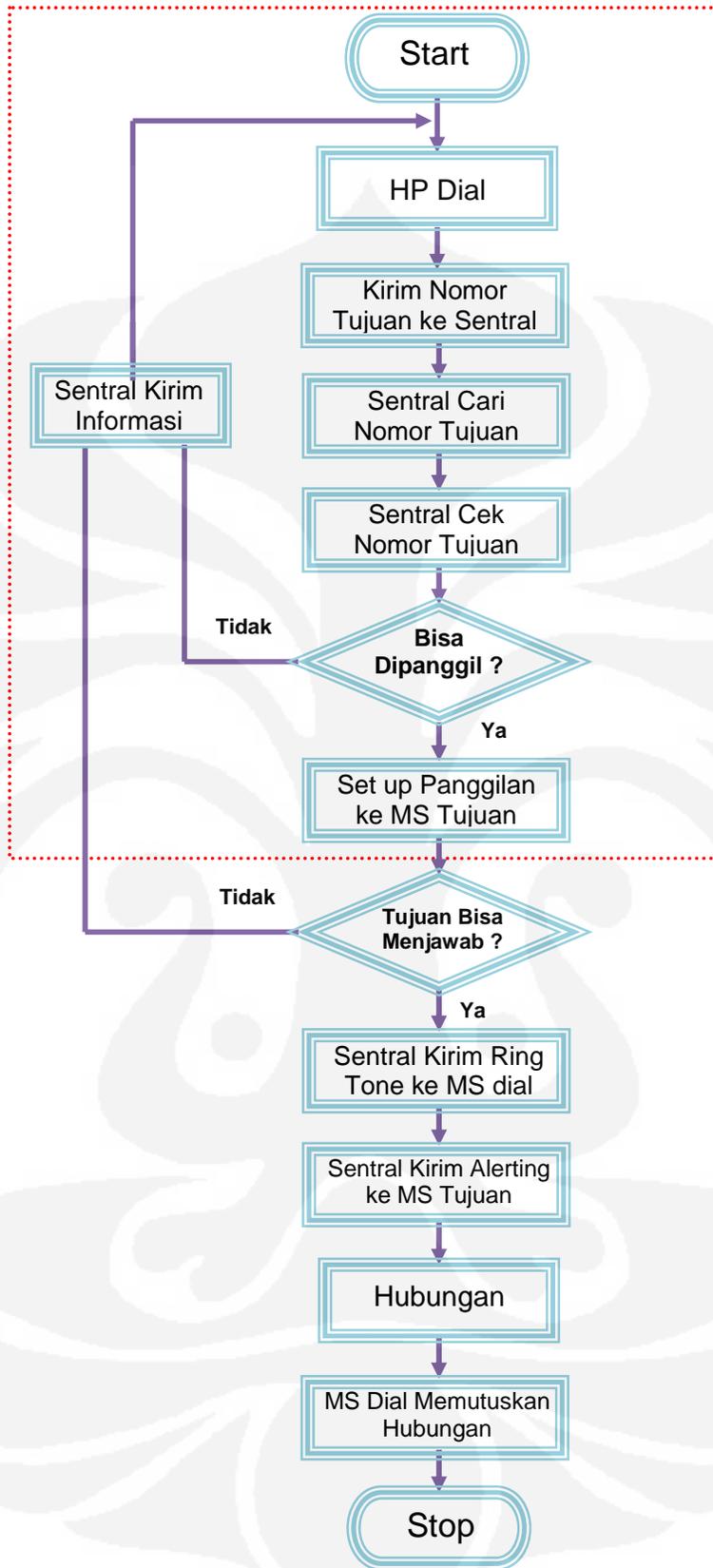
Kondisi pertama yang terjadi saat pensinyalan terjadi pada bagian *subscriber* pengirim adalah kondisi *idle*. Seperti telah dijelaskan pada bab dasar teori, saat MS berada dalam kondisi *idle*, terjadi beberapa tindakan dari MS yang memungkinkan fungsi pensinyalan tetap terjadi antara MS dengan MSC. Tindakan MS tersebut adalah identifikasi (*IMSI Attach*), *cell reselection*, dan *location area update*. Berdasarkan *message flow* dari ketiga tindakan MS saat kondisi *idle* tersebut, dapat dibuat sebuah *flow chart* yang merepresentasikan kondisi pensinyalan saat *idle*. Gambar 3.1 merupakan *flow chart* pensinyalan saat kondisi *idle* dengan tiga tindakan MS.



Gambar 3.1 Flow chart pensinyalan kondisi idle

Gambar 3.1 diatas merupakan *flow chart* saat kondisi *idle*. *Flow Chart* yang diberi kotak dengan garis putus-putus berwarna merah merupakan prosedur penentuan BTS yang *servicing* ke MS tersebut, atau disebut *IMSI attach*. Kemudian *flow chart* yang diberi kotak dengan garis putus-putus berwarna hijau adalah prosedur identifikasi ME yang digunakan. Sedangkan *flow chart* yang berada dalam kotak dengan garis putus-putus berwarna oranye adalah prosedur *update* lokasi.

Kondisi kedua yang terjadi saat pensinyalan terjadi pada bagian *subscriber* pengirim adalah terjadinya pengiriman informasi, dalam hal ini dibatasi pada saat MS melakukan panggilan keluar. Namun untuk memudahkan pemahaman mengenai konsep panggilan keluar yang terjadi pada *subscriber* pengirim, berikut ini diberikan *flow chart* yang menggambarkan proses komunikasi secara utuh pada *subscriber*. Gambar 3.2 menggambarkan *flow chart* proses komunikasi secara utuh pada *subscriber*.



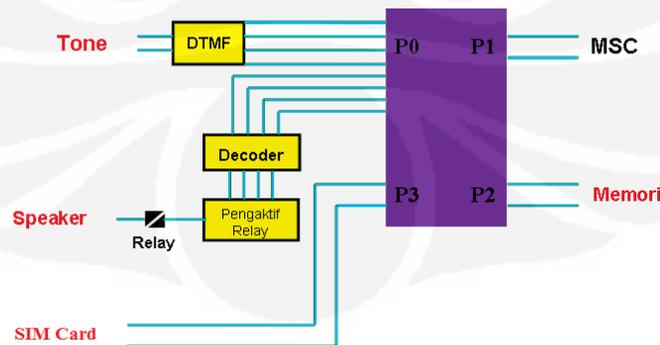
Gambar 3.2 Flow chart pensinyalan proses komunikasi utuh pada subscriber

Gambar 3.2 merupakan *flow chart* untuk panggilan pada komunikasi seluler secara keseluruhan. Gambar yang berada dalam kotak dengan garis putus-putus berwarna merah menunjukkan proses panggilan keluar, sedangkan yang berada di luar kotak dengan garis putus-putus berwarna merah menunjukkan proses panggilan masuk.

Pada *flow chart* panggilan keluar terlihat urutan proses yang terjadi saat MS melakukan panggilan keluar. Hal yang dilakukan pertama kali adalah MS melakukan proses *dial* dan memasukkan nomor tujuan. Selanjutnya nomor tujuan yang telah dimasukkan akan dikirimkan ke sentral. Sentral akan mengecek apakah nomor tujuan valid atau tidak dan juga mengecek apakah sibuk atau tidak. Pada saat proses ini terjadi MS yang memanggil akan menunggu selama kira-kira 5 detik. Setelah proses pengecekan selesai dan ternyata nomor tujuan valid dan tidak sibuk maka sentral akan membangun koneksi antara MS pemanggil dan tujuan. Setelah koneksi terbentuk maka komunikasi dapat berlangsung. Namun apabila nomor yang dipanggil sibuk, maka sentral akan mengirim berita kepada MS pemanggil bahwa nomor tujuan sibuk.

3.1.1.2 Algoritma Pensinyalan

Setelah *flow chart* pensinyalan baik kondisi *idle* maupun kondisi saat MS melakukan panggilan keluar selesai dibuat, selanjutnya adalah menyusun algoritma pensinyalan. Dalam penyusunan algoritma pensinyalan, terlebih dahulu dibuat blok diagram yang menggambarkan secara sederhana *chip* dan *port* pada sistem minimum mikrokontroler AT89S51. Gambar 3.3 merupakan blok diagram sederhana *chip* dan *port* dari sistem minimum mikrokontroler AT89S51.



Gambar 3.3 Blok diagram chip dan port sistem minimum mikrokontroler AT89S51

Pada Gambar 3.3 diatas, P0 di-*set* sebagai jalan dari tombol MS dan ke speaker untuk pensinyalan, P1 di-*set* sebagai jalan menuju MSC, P2 di-*set* sebagai jalan menuju Memori Internal, sedangkan P3 di-*set* sebagai jalan dari SIM Card.

Blok diagram pada Gambar 3.3 tersebut dibuat berdasarkan kebutuhan dalam pembuatan algoritma pensinyalan. Algoritma yang telah dibuat, digunakan untuk membuat program-program pensinyalan yang akan dihubungkan dengan mikrokontroler AT89S51.

Pada SIM Card didefinisikan alamat-alamat memori seperti diperlihatkan pada Tabel 3.1. Isi dari alamat memori ini adalah data-data yang terdapat pada SIM card.

Tabel 3.1. Tabel Definisi Alamat pada SIM Card

NO	Kode Desimal	Kode Biner	Isi Data
1	00	0000 0000	IMEI
2	01	0000 0001	LAI
3	02	0000 0010	Data Pelanggan
4	03	0000 0011	TMSI
5	04	0000 0100	IMSI
6	05	0000 0101	MSISDN
7	06	0000 0110	PIN
8	07	0000 0111	PUK
9	08	0000 1000	RAND
10	09	0000 1001	Chipering

Sedangkan pada memori internal di definisikan pada Tabel 3.2 berikut. Memori internal pada tabel 3.2 berikut ini adalah memori yang terdapat pada *mobile equipment*.

Tabel 3.2. Tabel Definisi Alamat pada Memori Internal

NO	Alamat memori	Isi Data
1	a	Bangun kanal pensinyalan
2	b	Pesan konfirmasi panggilan

3	c	Laporan alokasi TMSI selesai
4	d	Laporan <i>call release</i> selesai
5	e	Permintaan <i>update</i> lokasi

Pada tabel 3.1 dan 3.2 diperlihatkan isi dari alamat memori SIM Card dan isi dari alamat memori internal. Untuk memudahkan dalam pembuatan algoritma pensinyalan, pendefinisian kode biner alamat memori SIM Card dan alamat memori internal dilakukan dengan permisalan (tidak disesuaikan dengan sistem pensinyalan secara umum di Indonesia). Namun dalam pembuatan program-program pensinyalan, bit-bit yang tidak sesuai telah disesuaikan dengan bit-bit yang digunakan pada sistem pensinyalan komunikasi seluler yang umum digunakan di Indonesia.

Berdasarkan Gambar 3.1 dan 3.2 yaitu gambar *flow chart* pensinyalan kondisi *idle* dan gambar *flow chart* pensinyalan proses komunikasi utuh pada *subscriber*, dibuat algoritma pensinyalan untuk kondisi *idle* dan kondisi panggilan keluar. Untuk kondisi *idle*, terdapat tiga tahap utama pensinyalan yaitu IMSI *attach*, identifikasi ME (*Mobile Equipment*), dan *update* lokasi MS. Algoritma lengkap untuk pensinyalan pada kondisi *idle* dapat dilihat pada lampiran 1. Sedangkan untuk pensinyalan saat panggilan keluar, algoritma lengkapnya dapat dilihat pada lampiran 2.

3.1.1.3 Listing Program Pensinyalan

Setelah *flow chart* dan algoritma pensinyalan selesai dibuat, langkah selanjutnya adalah mengubah *flow chart* dan algoritma pensinyalan tersebut menjadi listing program pensinyalan dalam bahasa *assembly*. Berikut ini akan dijelaskan program pensinyalan dalam bahasa *assembly*.

3.1.1.3.1 IMSI Attach

Berdasarkan *flow chart* kondisi *idle* bagian IMSI *Attach* dan gambar 3.3, dapat dibuat sebuah algoritma sebagai berikut.

Label1 baca alamat 04 ; 04 adalah IMSI, P3 adalah port SIM Card
 kirim P3 ke A ; A adalah akumulator
 kirim A ke P1

hapus A
baca P1 ; P1 adalah sentral, isinya:TMSI
 kirim P1 ke A
 jika A = 0, kembali ke label1
hapus alamat 03 ; 03 tempat menyimpan TMSI
simpan A ke alamat 03
 kirim data alamat 03 ke A
 kirim A ke P3
hapus A

pada algoritma diatas, dijelaskan mengenai mekanisme IMSI Attach. Proses dimulai ketika alamat 04 telah terisi dengan IMSI. Kemudian IMSI tersebut dikirim ke *port 3* (P3) yang merupakan *sim card* tempat IMSI disimpan. Langkah selanjutnya adalah mengirim IMSI tersebut dari P3 ke P1, dimana P1 adalah *port* yang didefinisikan sebagai sentral. Proses ini dilanjutkan dengan penggantian IMSI dengan TMSI yang dilakukan oleh P1 sebagai sentral dan akan disimpan di alamat 03. Program bahasa *assembly* untuk algoritma diatas dapat dilihat pada lampiran 3 baris 43 sampai dengan 76

3.1.1.3.2 Identifikasi Mobile Equipment

Pada tahap ini, sentral, dalam hal ini port 1, melakukan identifikasi IMEI pada *Mobile Equipment*. Algoritmanya adalah sebagai berikut.

Label1 baca P1 ; P1 sentral, isinya permintaan ID ME
 kirim P1 ke A
 jika A=0, kembali ke label1
 kirim data alamat 00 ke A ; 00 adalah IMEI
 kirim A ke P1
hapus A
baca P1
 kirim P1 ke A
 jika A ≠0, kembali ke label1
selesai

Pada algoritma diatas, P1 berisi permintaan identifikasi ME, sementara alamat 00 berisi IMEI. Langkah awalnya adalah sentral mengirimkan permintaan untuk mengidentifikasi ME. Kemudian sebagai respon atas permintaan tersebut, MS mengirimkan IMEI yang menyatakan bahwa MS tersebut memiliki validitas yang benar dan dapat diproses ke tahap pensinyalan berikutnya. Program bahasa

assembly untuk algoritma identifikasi *Mobile Equipment* dapat dilihat pada lampiran 3 baris 78 sampai dengan 91

3.1.1.3.3 Update Lokasi

Tahap terakhir dari pensinyalan saat kondisi *idle* adalah *update* lokasi. Tahap ini diperlukan bila MS berpindah-pindah dari satu BTS ke BTS yang lainnya atau biasa dikenal dengan mekanisme *hand over*. Berikut ini adalah algoritmanya.

```
Label1 baca P1 ; P1 sentral
      kirim P1 ke A
      jika A ≠ data alamat 01, hapus A ; 01 adalah LAI
label2 kirim data alamat e ke A ; e adalah permintaan update lokasi
      kirim A ke P1
      hapus A
      baca P1
      kirim P1 ke A
      jika A=0, kembali ke label1
      hapus data alamat 01
      simpan A ke alamat 01
      hapus A
      kirim data alamat 01 ke A
      jika A=0, hapus A
      kembali ke label2
      kirim data alamat c ke P1 ; c adalah laporan alokasi TMSI selesai
      baca P1
      kirim P1 ke A
      jika A=0, kembali ke label2
selesai
```

Tahap *update* lokasi diawali dengan pengecekan LAI (*Location Area Identity*). Bila LAI tidak sesuai dengan sentral, maka dilakukan permintaan *update* lokasi oleh MS. Identitas yang dikirim sebagai respon atas permintaan *update* lokasi adalah TMSI yang merupakan nomor sementara yang digunakan pelanggan saat berpindah area. Setelah proses pemberian TMSI dari sentral (P1) ke MS selesai, maka selesailah tahap terakhir dari pensinyalan kondisi *idle*. Program *assembly* untuk *update* lokasi dapat dilihat pada lampiran 3 baris 93 sampai dengan 135

3.1.1.3.4 Proses panggilan keluar pada MS

Setelah algoritma yang mendasari pembuatan program *assembly* pensinyalan untuk kondisi *idle* selesai dibahas, berikut ini akan dijelaskan algoritma yang mendasari pembuatan program *assembly* pensinyalan saat *Mobile Station* selaku *subscriber* melakukan panggilan keluar. Algoritma adalah sebagai berikut.

Label1 baca data alamat *a* ; *a* adalah bangun kanal pensinyalan
P0 adalah MS

Kirim P0 ke A

Kirim A ke P1

Hapus A

Label2 Baca *P1* ; persetujuan bangun kanal

Kirim P1 ke A

Jika A=0, Kembali ke label 1

Label3 baca data alamat *04* ; *04* adalah IMSI, *P3* adalah SIM Card

Kirim P3 ke A

Kirim A ke P1

Hapus A

Baca P1

Kirim P1 ke A

Jika A=0, Kembali ke label3

Label4 baca *P1* ; *P1* sentral, berisi permintaan autentikasi

Kirim P1 ke A

Jika A=0, Kembali ke label4

Kirim A ke P0

Hapus A

Label5 Baca *P0*

Kirim P0 ke A

Jika A=0, Kembali ke label5

Hapus A

Label6 baca data alamat *08* ; *08* adalah RAND, *P3* adalah SIM Card

Kirim P3 ke A

Kirim A ke P1

Hapus A

Baca P1

Kirim P1 ke A

Jika A=0, Kembali ke label6

Label7 baca *P1* ; *P1* sentral, berisi permintaan *chipering*

Kirim P1 ke A

Jika A=0, Kembali ke label7

Kirim A ke P0

Hapus A
Label8 baca P0
Kirim P0 ke A
Jika A=0, Kembali ke label8
Hapus A
Label9 baca data alamat 09 ; 09 adalah *chipering*, P3 adalah *SIM Card*
Kirim P3 ke A
Kirim A ke P1
Hapus A
Baca P1
Kirim P1 ke A
Jika A=0, Kembali ke label9

Label10Baca P1 ; pesan berupa pesan *set up* panggilan
Kirim P1 ke A
Jika A=0, Kembali ke label2
Simpan A ke alamat 03 ; 03 adalah TMSI
Hapus A
Baca P3 ; persetujuan panggilan
Kirim P3 ke A
Jika A=0, Kembali ke label10
selesai

Proses ini diawali dengan pembentukan kanal pensinyalan yang akan membatasi koneksi antara MS (P0) dengan MSC (P1). Permintaan pembentukan kanal dilakukan oleh MS, dalam hal ini *port 0*, kepada MSC. Setelah MSC menyetujui pembentukan kanal P1 memberikan respon pada MS. Langkah berikutnya adalah pengiriman IMSI dari *port 3* (P3) selaku *SIM Card* ke sentral (P1). Setelah pengiriman IMSI selesai, sentral meminta autentikasi pada MS untuk dapat meneruskan ke proses *set up* panggilan. Kode-kode autentikasi terdapat pada P3 yaitu *SIM Card*. Untuk merespon permintaan autentikasi, P3 mengirimkan kode-kode autentikasi yang terdapat pada alamat 08 dan 09 ke P1 selaku sentral. Saat proses ini selesai, P1 akan mengirim pesan untuk *set up* panggilan yang akan disetujui oleh MS dengan mengirimkan TMSI.

Dalam penerjemahan algoritma pensinyalan ini menjadi program, terdapat kode-kode yang tidak dicantumkan yaitu kode-kode autentikasi. Hal ini dikarenakan kode-kode ini hanya terdapat dalam algoritma khusus yang hanya dimiliki oleh perusahaan-perusahaan telekomunikasi yang memiliki sistem pensinyalan untuk komunikasi seluler. Program *assembly* pensinyalan untuk

kondisi panggilan keluar ini dapat dilihat pada lampiran 3 baris 137 sampai dengan 172

3.1.1.3.5 Program Sentral

Dalam perancangan program sentral untuk rangkaian simulator system pensinyalan komunikasi seluler bagian *subscriber* pengirim, digunakan prosedur seperti sebuah saluran informasi dimana di dalamnya akan terjadi aliran data yang menunjukkan permintaan dari MS pengirim dengan respon dari sentral. Dalam program yang dibuat, saluran informasi dibuat dalam blok-blok yang terdiri dari 10 blok dengan alamat memori yang digunakan adalah dari 60h sampai dengan 69h. blok-blok informasi ini nantinya akan diisi dengan data-data yang berupa permintaan dari MS pengirim dan akan dibalas dengan respon dari sentral. Untuk memudahkan pembuatan prosedur ini, digunakan dua buah register yang akan dianalogikan sebagai saat mengirim dan saat membaca data. Register yang digunakan untuk mengirim data adalah register 5 (R5), sedangkan register yang digunakan untuk membaca data adalah register 4 (R4).

Prosedur sentral ini akan dimulai bila ada data yang masuk ke R5. Data yang masuk akan menempati blok-blok yang kosong. Nilai R5 dan R4 saat data belum masuk adalah sama. Sedangkan bila sudah ada data yang masuk, nilai R4 akan lebih kecil 1 nilai dari R5 karena R5 lebih dulu diakses untuk mengirim data. Proses ini akan selesai saat nilai R5 dan R4 menunjukkan angka yang sama. Hal itu tersebut berarti semua data yang masuk ke saluran informasi yang merupakan permintaan dari MS pengirim kepada sentral telah dibaca dan diberikan respon oleh sentral.

3.1.1.3.6 Program Delay

Pada rancang bangun ini, program *delay* digunakan untuk memberikan jeda saat menampilkan nilai-nilai penting pensinyalan pada rangkaian *seven segment*. Program *delay* yang dibuat dapat diubah-ubah nilainya mulai dari *delay* 1 detik dan seterusnya. Dalam perancangan program ini, waktu *delay* yang digunakan adalah 6 detik.

3.1.2 Rangkaian Mikrokontroller untuk Simulator Pensinyalan

Dalam membuat rangkaian mikrokontroller untuk simulator pensinyalan, terdapat tiga bagian rangkaian yang membentuk kesatuan untuk simulator pensinyalan. Bagian-bagian tersebut adalah sistem minimum AT89S51, dalam hal ini adalah DT-51 *Low Cost Micro System Ver 2.0* yang merupakan produk dari *innovative electronics*, rangkaian tombol input yang direpresentasikan sebagai *keypad* pada *Mobile Station*, dan rangkaian *seven segment* yang direpresentasikan sebagai layar pada *Mobile Station* serta untuk menampilkan bit-bit *message type* dari *message flow* pensinyalan. Berikut ini adalah penjelasan berdasarkan bagian-bagian rangkaian yang membentuk rangkaian mikrokontroller untuk simulator pensinyalan.

3.1.2.1 Sistem Minimum AT89S51

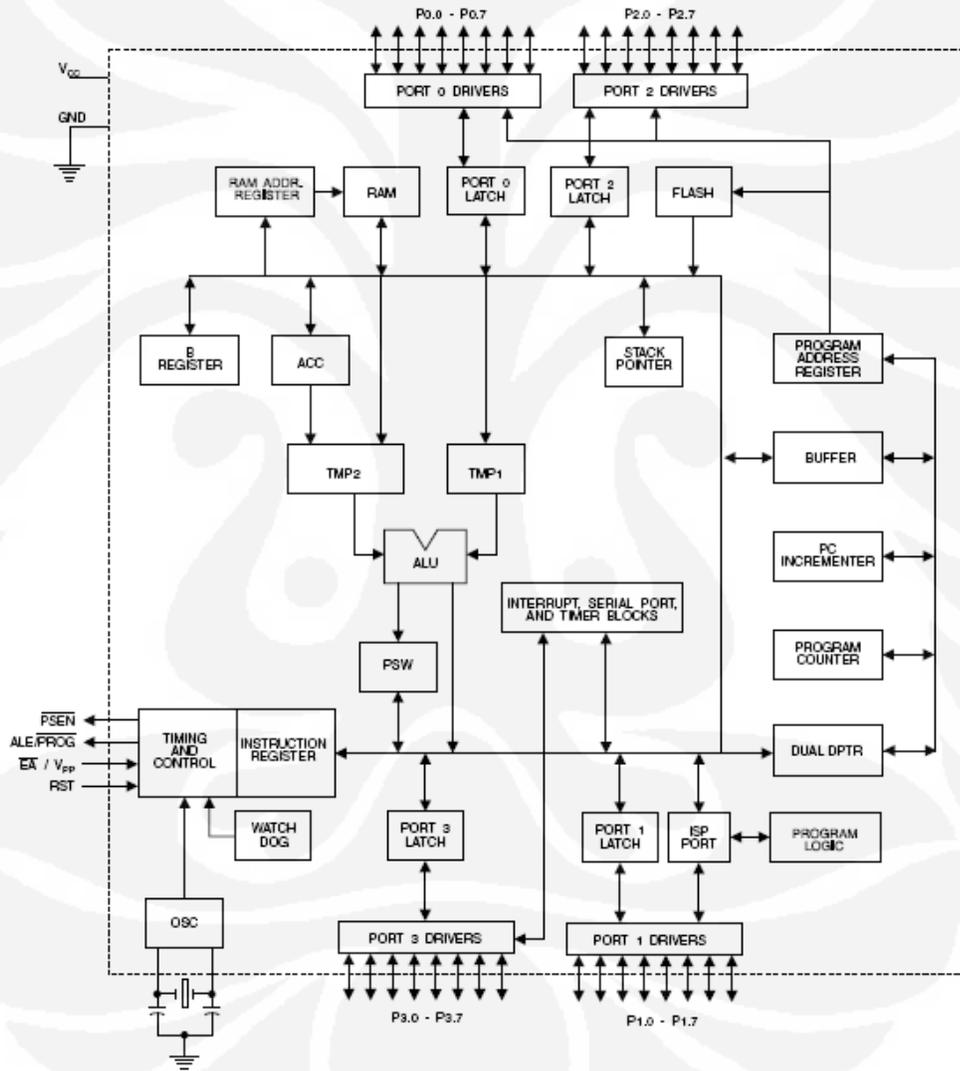
Dalam perancangan rangkaian simulator pensinyalan komunikasi seluler dengan mikrokontroller AT89S51, sistem minimum AT89S51 berfungsi sebagai “otak” dari sistem, yang artinya sistem minimum AT89S51 menjadi bagian utama dari rangkaian simulator pensinyalan komunikasi seluler. Pada sistem minimum ini, terdapat empat *port* yang masing-masing akan digunakan sebagai berikut :

- ❖ *Port 0* : untuk keperluan menampilkan data-data penting, dalam hal ini adalah data-data yang memuat bit-bit yang dikirimkan ke sentral selama proses pensinyalan berlangsung. Bit-bit data ini merupakan bit-bit data yang valid, karena didapatkan langsung dari salah satu *vendor* telekomunikasi di Indonesia. Rangkaian yang digunakan untuk menampilkan data-data ini adalah rangkaian *seven segment*.
- ❖ *Port 1* : fungsi dari *port 1* sama dengan *port 0* yaitu untuk menampilkan data-data penting. Perbedaannya terletak pada nilai dari bit decimal yang ditampilkan oleh *port 1* bernilai puluhan dan satuan, sedangkan nilai dari bit decimal yang ditampilkan oleh *port 0* bernilai ratusan. Untuk lebih jelas mengenai hal ini dapat dilihat pada sub bab 3.1.2.3
- ❖ *Port 2* : untuk keperluan input, dalam hal ini adalah nomor-nomor yang dianalogikan sebagai nomor-nomor *Mobile Station* tujuan. Rangkaian

yang digunakan pada *port 2* ini adalah rangkaian *keypad*. Untuk lebih jelas mengenai rangkaian *keypad* yang digunakan sebagai input, dapat dilihat pada sub bab 3.1.2.2

❖ *Port 3* : sebagai sentral (*Mobile Switching Center*)

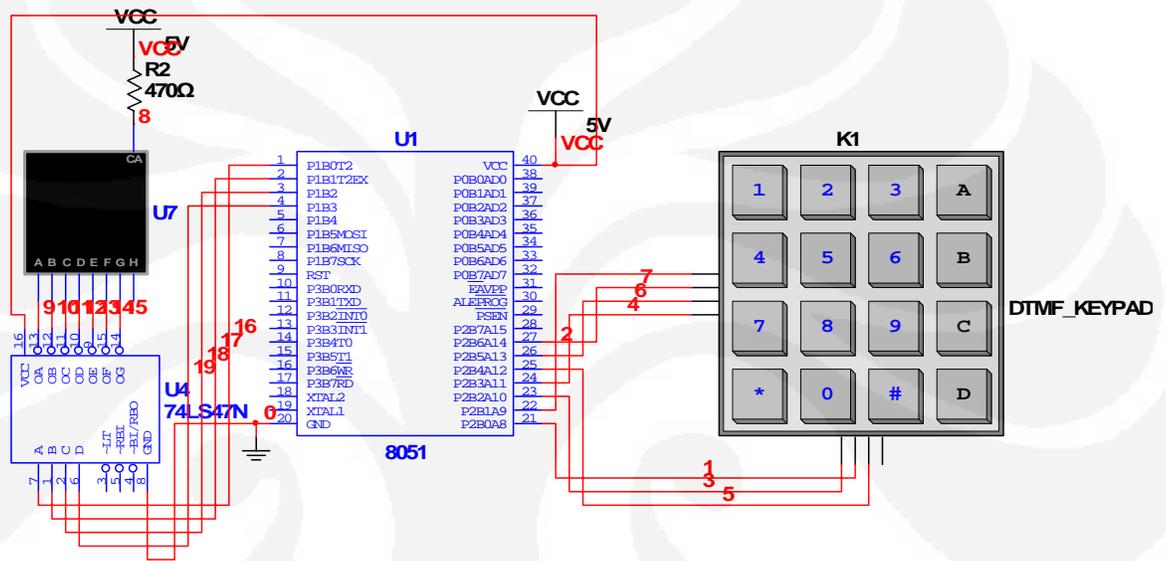
Gambar 3.4 menunjukkan rangkaian sistem minimum mikrokontroler AT89S51 yang digunakan sebagai bagian utama dari rangkaian simulator pensinyalan komunikasi seluler.



Gambar 3.4 Rangkaian Sistem Minimum AT89S51 [11]

3.1.2.2 Rangkaian Keypad

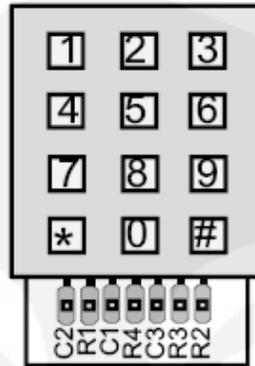
Gambar 3.5 memperlihatkan rangkaian *keypad* yang dihubungkan dengan *chip* mikrokontroler AT89S51. Pada gambar terlihat bahwa yang digunakan adalah *DTMF keypad* dan bukan *numeric keypad*. Hal ini dikarenakan penggunaan *keypad* dalam rangkaian simulator pensinyalan komunikasi seluler adalah sebagai input nomor dan bukan sebagai alat hitung.



Gambar 3.5 Rangkaian Keypad

Prinsip kerja *keypad* ini didasarkan pada prinsip kerja matrik, yaitu dalam hal ini yang terdiri dari tiga kolom dan empat baris. Pada Gambar 3.5 terlihat bahwa kolom *keypad* ada empat, namun yang digunakan untuk input simulator hanya tiga kolom yaitu yang berisi angka-angka saja. Prinsip kerja matriks yang dimaksud adalah kombinasi logika dari baris dan kolom *keypad*. *Keypad* ini bekerja berdasarkan *active low*, yang artinya ketika menekan satu tombol pada *keypad*, maka kolom dan baris yang berhubungan akan bernilai logika “0”. sebagai contoh, bila angka “3” pada *keypad* ditekan, maka kolom tiga dan baris satu akan memberikan nilai “0” pada *port 2* pin 4 dan *port 2* pin 1, sedangkan pada *port 2* pin yang lain akan bernilai “1”. Dalam penyusunan sambungan antara

port 2 dengan pin-pin kolom dan baris *keypad*, dilakukan berdasarkan skema dasar dari pin-pin kolom dan baris *keypad* seperti pada Gambar 3.6 berikut ini.



Gambar 3.6 Skema Konfigurasi Pin Keypad

Pada Gambar 3.6 diperlihatkan skema konfigurasi pin *keypad* yang terdiri dari kolom dan baris (dilambangkan dengan “C” untuk kolom dan “R” untuk baris) ketujuh pin inilah yang dihubungkan dengan *port 2* pin 0 sampai dengan pin 6 pada *chip* AT89S51. Penyambungan pin *keypad* dimulai dari kiri ke kanan (dimulai dari C2 sampai dengan R2) dipasang dengan pin *port 2* dengan nilai yang paling kecil, yaitu pin 0. Sehingga C2 dihubungkan dengan pin 0 *port 2* dan seterusnya sampai dengan R2 dihubungkan dengan pin 6 *port 2*. berikut ini adalah tabel yang menunjukkan representasi bit-bit pada *port 2* bila angka-angka pada *keypad* ditekan.

Tabel 3.3. Tabel representasi bit *port 2* saat tombol *keypad* ditekan

Tombol <i>Port</i>	P 2.0	P 2.1	P 2.2	P 2.3	P 2.4	P 2.5	P 2.6
Pin Keypad	C2	R1	C1	R4	C3	R3	R2
0	0	1	1	0	1	1	1
1	1	0	0	1	1	1	1
2	0	0	1	1	1	1	1
3	1	0	1	1	0	1	1
4	1	1	0	1	1	1	0

5	0	1	1	1	1	1	0
6	1	1	1	1	0	1	0
7	1	1	0	1	1	0	1
8	0	1	1	1	1	0	1
9	1	1	1	1	0	0	1
*	1	1	0	0	1	1	1
#	1	1	1	0	0	1	1

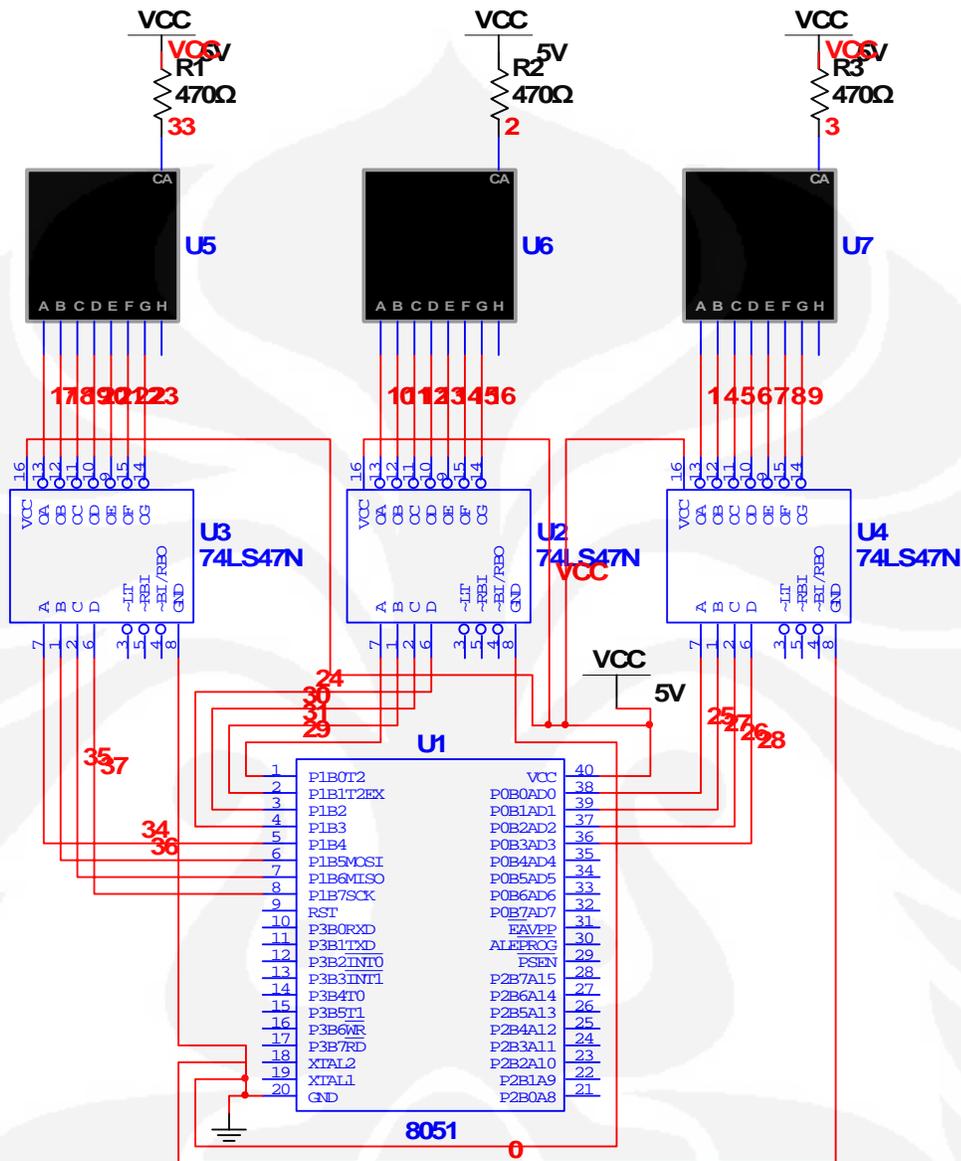
Pada tabel 3.3 diatas, dengan jelas diperlihatkan nilai bit biner yang akan muncul pada *port 2* bila salah satu angka pada *keypad* ditekan. Terlihat bahwa mekanisme kerja *keypad* adalah *active low*, yang artinya bit biner bernilai "0" menunjukkan tombol yang ditekan pada kolom dan baris yang bersesuaian. Hasil penekanan tombol *keypad* akan ditampilkan pada rangkaian *seven segment*.

3.1.2.3 Rangkaian seven segment

Gambar 3.7 menunjukkan rangkaian *seven segment* yang digunakan untuk menampilkan data-data penting berupa bit-bit *message type* yang didapatkan dari salah satu *vendor* telekomunikasi di Indonesia. Rangkaian *seven segment* ini akan dihubungkan dengan *port 0* dan *port 1* dari sistem minimum AT89S51.

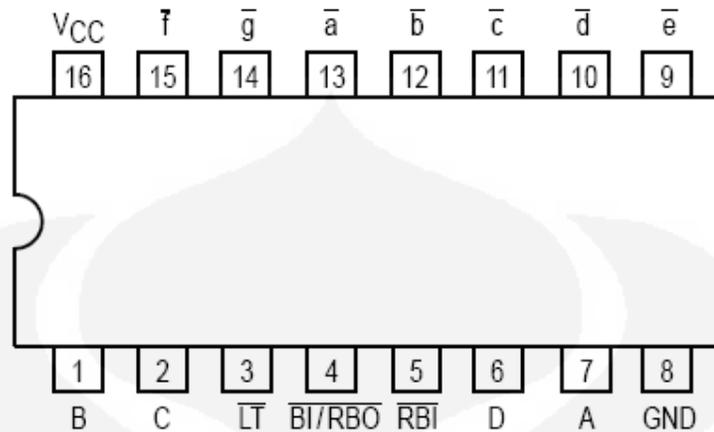
Perancangan dan pembuatan rangkaian ini dilakukan dengan menggunakan PCB *board* berlubang, tiga buah *seven segment common anoda*, tiga buah *resistor* 470 Ω , tiga buah *decoder* 74LS47N, dan *header*. Berikut ini adalah langkah-langkah yang dilakukan dalam proses pembuatan rangkaian *seven segment*.

1. Menyusun tiga buah *seven segment*, tiga buah *decoder* 74LS47N, tiga buah *resistor* 470 Ω dan dua buah *header*, masing-masing dengan 10 pin, berdasarkan Gambar 3.6. kemudian susunan ini diletakkan pada PCB *board* berlubang.
2. Menyambungkan komponen-komponen yang telah disolder dengan menggunakan kabel tembaga serabut. Penyambungan komponen rangkaian berdasarkan Gambar 3.7.



Gambar 3.7 Rangkaian Seven Segment

Pada Gambar 3.7 yang merupakan rangkaian *seven segment*, terlihat bahwa dari tiga buah *seven segment* yang digunakan, satu *seven segment* dihubungkan dengan *port 0*, sedangkan dua buah *seven segment* dihubungkan dengan *port 1*. Susunan ini dibuat berdasarkan program pensinyalan yang dibuat berdasarkan prinsip kerja *BCD to Seven Segment Decoder* yang digunakan adalah *decoder 74LS47N* yaitu jenis *BCD to seven segment decoder* yang memiliki *connection diagram* sebagai berikut.



Gambar 3.8 Connection Diagram BCD to 7-segment Decoder 74LS47N

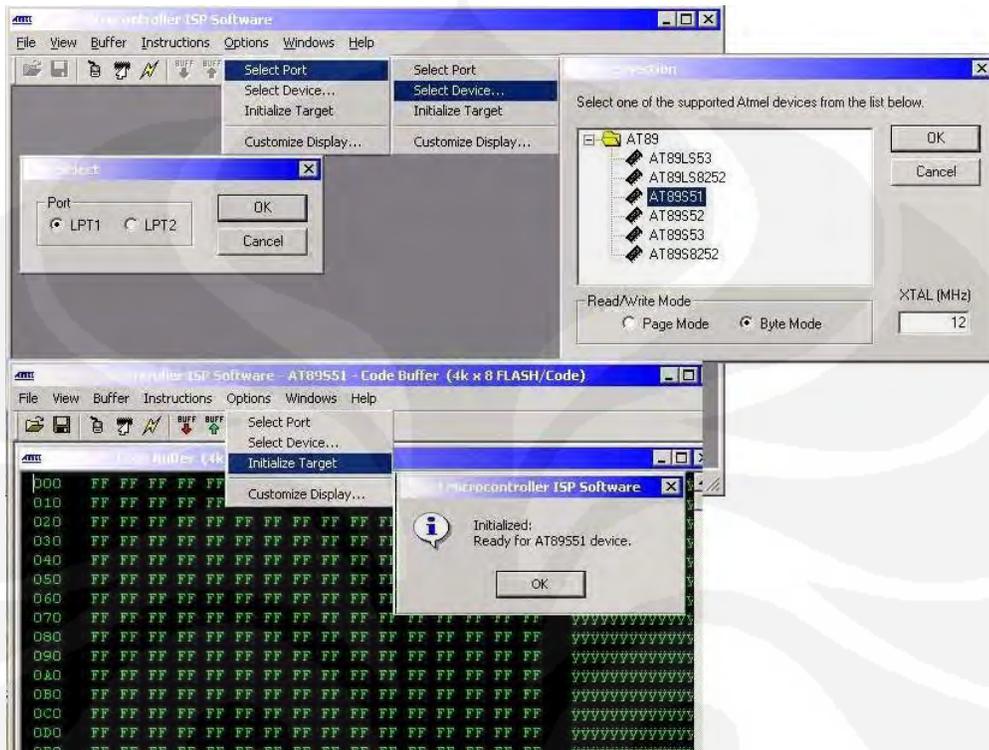
Pada Gambar 3.8 terlihat bahwa *decoder 74LS47N* memiliki pin-pin a, b, c, d, e, f, g, yang sejajar dengan pin Vcc. Pin-pin itulah yang disambungkan dengan pin-pin a, b, c, d, e, f, g, yang terdapat pada *seven segment*. Sedangkan pada bagian yang lain, terdapat pin-pin A, B, C, D, yang sejajar dengan pin GND. Pin-pin tersebut akan dihubungkan dengan *port 0* dan *port 1* pada sistem minimum AT89S51. Keadaan yang terjadi saat rangkaian *seven segment* dihubungkan dengan sistem minimum AT89S51 serta analisa hasil keluaran pada *seven segment*, dapat dilihat pada bab 4.

3.1.2.4 Proses Burning

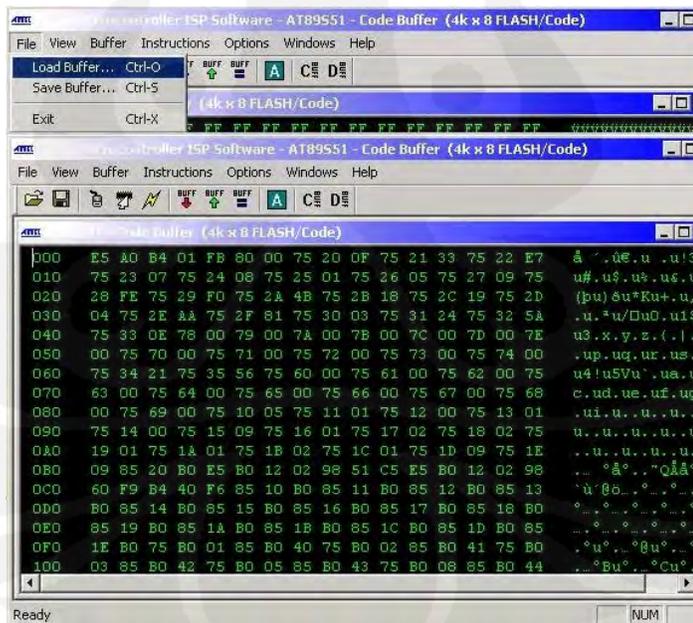
Proses *burning* dilakukan untuk memasukkan program pensinyalan yang telah dibuat ke dalam mikrokontroller AT89S51. *Software* yang digunakan untuk proses *burning* ini adalah *Atmel microcontroller in-system programming (ISP)*. data program yang dimasukkan saat proses *burning* ini adalah data program dengan jenis *file .HEX* yang didapatkan dengan meng-*compile* program yang telah dibuat menggunakan *software 8051IDE*. Berikut ini adalah tahap-tahap yang dilakukan saat proses *burning*:

1. Menghubungkan *port burning* system minimum AT89S51 ke computer dengan menggunakan kabel parallel.
2. Melakukan inisialisasi.
3. Melakukan *load program*.
4. *Burning*.

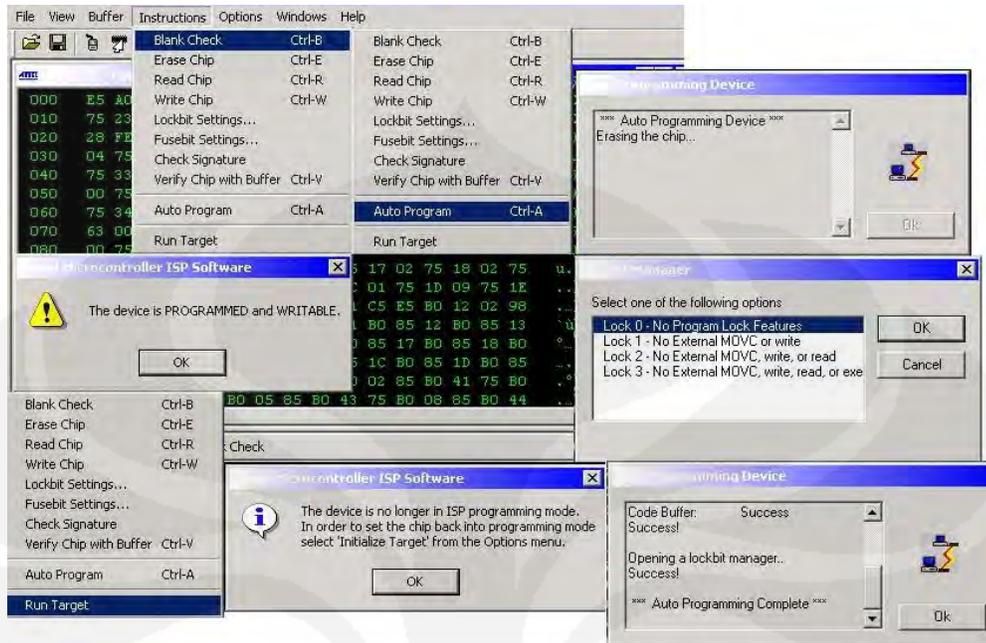
Berikut ini adalah gambar-gambar yang memperlihatkan tahapan proses *burning*:



Gambar 3.9 Tahap Inisialisasi



Gambar 3.10 Tahap Load Program



Gambar 3.11 Tahap Burning

Gambar 3.9, 3.10. dan 3.11 masing-masing menjelaskan langkah-langkah yang perlu dilakukan untuk masing-masing tahapan. Secara umum, semua tahapan tersebut dapat dirangkum menjadi:

❖ **Tahap Inisialisasi**

Selec port ➡ LPT1 ➡ *Select Device* ➡ AT89S51 ➡ *Initialize Target*

❖ **Tahap Load Program**

Load Buffer ➡ Mencari *file* yang diinginkan

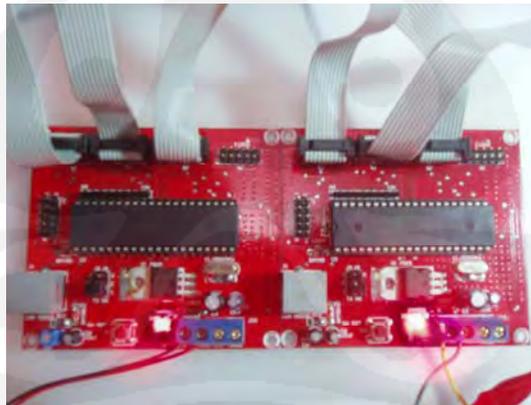
❖ **Tahap Burning**

Blank Check ➡ *Auto Program* ➡ *Run Target*

Pada Gambar 4.1, diperlihatkan rangkaian keseluruhan dari simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim. Tampak pada gambar masing-masing sub-sistem yang menyusun rangkaian simulator pensinyalan yaitu rangkaian sistem minimum mikrokontroler AT89S51, rangkaian *keypad* sebagai input, dan rangkaian *seven segment* sebagai output yang menampilkan nilai-nilai yang diinginkan. Untuk mengetahui lebih jelas performa dari masing-masing sub sistem yang menyusun rangkaian simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim ini, berikut ini adalah penjelasannya.

4.1.1 Sub-sistem Rangkaian Sistem Minimum AT89S51

Analisa untuk sub-sistem rangkaian sistem minimum AT89S51 ditentukan dengan melihat keluaran pada masing-masing *port* apakah sudah sesuai dengan hasil simulasi program *assembly* pada simulator 8051IDE. Pada rangkaian simulator pensinyalan ini, system minimum yang digunakan sebanyak dua buah. Yang pertama digunakan sebagai rangkaian untuk program utama. Sedangkan yang kedua digunakan sebagai rangkaian untuk program *keypad*. Berikut ini adalah gambar dua rangkaian system minimum yang telah terhubung satu sama lain.



Gambar 4.2 dua rangkaian system minimum yang telah terhubung

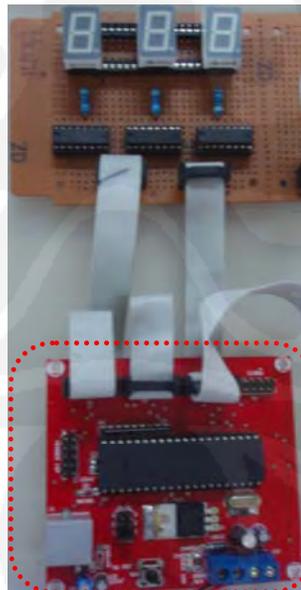
Pada Gambar 4.2 diatas, terlihat dua buah rangkaian system minimum AT89S51 yang telah terhubung satu sama lain. Rangkaian system minimum sebelah kiri selanjutnya dihubungkan dengan rangkaian *seven segment*, sedangkan rangkaian system minimum sebelah kanan dihubungkan dengan rangkaian *keypad*.

4.1.1.1 Rangkaian system minimum untuk program pensinyalan utama

Beberapa hal yang diperhatikan pada rangkaian sistem minimum ini adalah sebagai berikut :

1. *Port 0*, yang berfungsi sebagai *port* yang dihubungkan dengan *seven segment* dianalisa apakah keluarannya sudah sama dengan nilai-nilai yang akan ditampilkan di *seven segment* bagian nilai decimal ratusan.
2. *Port 1*, analisa yang dilakukan pada *port 1* juga sama dengan yang dilakukan pada *port 0*, yaitu melihat nilai keluaran kemudian dibandingkan dengan nilai yang akan ditampilkan di *seven segment*. Untuk *port 1*, nilai yang dilihat adalah nilai decimal puluhan dan ratusan.
3. *Port 2*, yang berfungsi sebagai *port* yang dihubungkan dengan rangkaian *keypad* dianalisa apakah keluarannya sudah sama dengan nilai tombol *keypad* yang ditekan.
4. *Chip* mikrokontroler AT89S51, yang merupakan bagian penting sebagai komponen pengendali, pengontrol, dan penyimpan data serta memori.

Gambar 4.3 berikut menjelaskan mengenai arsitektur pada rangkaian system minimum untuk program pensinyalan utama.



Gambar 4.3 rangkaian system minimum untuk program pensinyalan utama

Pengujian untuk masing-masing bagian ini dilakukan sebagai berikut:

a. Port 0

Port 0, yang terdiri dari sepuluh pin, yaitu pin 0 sampai dengan pin 9, dengan pin 0 sebagai *ground* dan pin 1 sebagai *Vcc*, dalam kesatuan rangkaian simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim berfungsi sebagai *port* yang dihubungkan dengan *seven segment* bagian nilai decimal ratusan. Nilai-nilai yang ditampilkan pada *seven segment* adalah bit-bit penting yang ada selama proses pensinyalan berlangsung dan nilai hasil penekanan tombol *keypad*. Saat program pensinyalan *assembly* disimulasikan dengan simulator 8051IDE supaya kode *mnemonic* program dapat dikenali oleh mikrokontroler, nilai-nilai yang ditampilkan di *seven segment* telah dikirimkan ke *port 0*. Pengecekan nilai-nilai pada pin-pin *port 0* dilakukan dengan menggunakan multimeter. Pengecekan dilakukan dengan mengecek satu persatu pin-pin tersebut untuk mengetahui ada atau tidaknya tegangan pada masing-masing pin tersebut. Bila ada tegangan, maka pin tersebut bernilai biner “1”, namun bila yang terjadi sebaliknya, maka pin tersebut bernilai “0”. Setelah delapan pin selesai dicek, yaitu pin 2 sampai dengan pin 9, maka nilai biner yang diketahui berdasarkan tegangan akan dicocokkan dengan nilai biner *port 0* yang dimasukkan pada program. Dari hasil pengecekan yang dilakukan, didapatkan bahwa semua nilai pin-pin *port 0* sudah memiliki nilai yang sama dengan nilai biner yang dikirimkan ke *port 0* pada program *assembly*.

b. Port 1

Unjuk kerja *port 1* pada sistem minimum AT89S51 dapat diketahui melalui cara yang sama dengan pengujian pada *port 0*, yaitu dengan cara pengecekan tegangan pada pin-pin *port 1* untuk mengetahui nilai biner yang dihasilkan. Karena *port 1* memiliki fungsi yang sama dengan *port 0* pada kesatuan rangkaian simulator pensinyalan komunikasi seluler, maka cara pengujian *port 1* juga sama dengan *port 0*, hanya saja *port 0* memiliki nilai keluaran decimal untuk ratusan, sedangkan *port 1* memiliki nilai keluaran decimal untuk puluhan dan satuan.

Pada program pensinyalan komunikasi seluler, bagian program yang digunakan untuk menampilkan nilai-nilai ke *seven segment* menggunakan suatu

prinsip kerja *BCD to seven segment* sehingga dari nilai biner yang dimasukkan ke dalam *port 0* akan diubah menjadi nilai decimal pada *seven segment* sehingga pada saat pengecekan pada pin, nilai yang menjadi acuan adalah nilai biner bukan nilai decimal. Dari hasil pengujian yang dilakukan, didapatkan bahwa semua nilai pin-pin *port 1* sudah memiliki nilai yang sama dengan nilai biner yang dikirimkan ke *port 1* pada program *assembly*.

c. Port 2

Uji performa *port 2* sebagai *port* yang dihubungkan dengan rangkaian *keypad* dapat diketahui dengan mengecek pin-pin *port 2* dengan menggunakan multitester. Pengecekan pin-pin *port 2* dilakukan untuk mengetahui apakah nilai biner yang muncul pada pin-pin *port 2* sama dengan nilai tombol *keypad* yang ditekan. Nilai biner diketahui bila pin *port 2* memiliki tegangan, yang artinya bernilai “1”, sedangkan bila tidak ada tegangan berarti bernilai “0”. Dari hasil pengujian yang dilakukan, didapatkan bahwa semua nilai pin-pin *port 2* sudah memiliki nilai yang sama dengan nilai biner yang dikirimkan ke *port 2* pada program *assembly*.

d. Chip Mikrokontroler AT89S51

Performa *chip* mikrokontroler AT89S51 dalam sistem minimum ini dapat diketahui dengan memasukkan program pensinyalan ke dalam *chip* ini melalui proses *burning* yang dilakukan menggunakan *Atmel microcontroller in-system programming (ISP) Software*, atau yang lebih dikenal sebagai *ISP software*. Setelah proses *burning* berhasil, maka dilanjutkan dengan proses *running*, yaitu menjalankan langkah-langkah instruksi program pensinyalan yang telah dimasukkan. Apabila hasil yang diinginkan pada program berhasil diperlihatkan pada rangkaian mikrokontroler, maka *chip* mikrokontroler memiliki performa yang baik.

Pada saat pengujian *chip* mikrokontroler AT89S51 dilakukan, beberapa kali ditemui kegagalan saat proses *burning* dan *running*, namun setelah diteliti ternyata kesalahan bukan terletak pada *chip* mikrokontroler AT89S51, akan tetapi pada kabel konektor yang menghubungkan PC (*Personal Computer*) dengan sistem minimum AT89S51. Selain itu kegagalan proses *burning* dan *running* juga

dikarenakan adaptor sebagai sumber tegangan sistem minimum beberapa kali mengalami kekenduran pada bagian yang dihubungkan dengan sistem minimum. Yang terakhir, kegagalan proses *running* dikarenakan konektor rangkaian-rangkaian yang tidak pas.

Dari hasil analisa diatas dan pengujian yang dilakukan, dapat disimpulkan bahwa *chip* mikrokontroler AT89S51 memiliki performa yang baik sebagai komponen pengendali, pengontrol, dan penyimpan data serta memori pada kesatuan rangkaian simulator pensinyalan komunikasi seluler.

4.1.1.2 Rangkaian system minimum untuk program keypad

Pengujian untuk system minimum ini dilakukan pada bagian-bagian sebagai berikut:

1. *Port 0*, dalam hal ini adalah *port* yang dihubungkan dengan *port 2* pada system minimum untuk program utama. *Port* ini berfungsi sebagai penghubung yang akan memberikan tanda apabila system minimum untuk program *keypad* telah melakukan proses *dial* (penekanan tombol bintang ‘*’)
2. *Port 1*, pada system minimum ini digunakan untuk menampilkan nilai tombol yang ditekan pada *keypad*.
3. *Port 2*, berfungsi sebagai *port* yang dihubungkan dengan *keypad*.
4. *Chip* mikrokontroler AT89S51, yang merupakan bagian penting sebagai komponen pengendali, pengontrol, dan penyimpan data serta memori.

Gambar 4.4 berikut menjelaskan mengenai arsitektur pada rangkaian system minimum untuk program pensinyalan utama.



Gambar 4.4 rangkaian system minimum untuk program keypad

Pengujian untuk masing-masing bagian system minimum ini dilakukan sebagai berikut:

a. Port 0

Pengujian pada *port 0* dilakukan untuk mengecek apakah nilai *port 0* sudah sesuai dengan nilai yang seharusnya muncul saat tombol bintang '*' pada *keypad* ditekan yang menandakan terjadinya proses *dial*. Pada listing program *keypad* yang dibuat, dapat dilihat pada lampiran 1, indikator yang menandakan bahwa proses panggilan telah dimulai adalah ditekannya tombol '*' pada *keypad*. Setelah tombol tersebut ditekan, maka nilai *port 0* seharusnya bernilai biner 0000 0001. Nilai tersebut telah dimasukkan pada program *keypad* untuk menandakan bahwa tombol bintang sebagai tombol *dial* telah ditekan.

Pengujian pin-pin *port 0* yang dilakukan dengan multimeter dengan cara yang sama untuk mengetes system minimum program utama, menunjukkan hasil yang sesuai dengan yang diharapkan, yaitu nilai pin-pin *port 0* menunjukkan nilai 0000 0001. Sehingga dapat dikatakan performa *port 0* pada system minimum program *keypad* berjalan baik.

b. Port 1

Port 1, yang dalam hal ini sebagai *port* yang dihubungkan dengan satu buah *seven segment*, merupakan *port* yang digunakan untuk menampilkan nilai tombol hasil penekanan pada *keypad*. Pengujian *port* ini dilakukan untuk mengetahui kesesuaian nilai yang ditampilkan pada *seven segment* dengan tombol yang ditekan pada *keypad*.

Pengujian dilakukan saat program *keypad* telah di-*burn* ke dalam mikrokontroler AT89S51. Setelah *burning* selesai, pengujian dilakukan dengan melakukan penekanan ke semua tombol *keypad* kecuali tombol bintang '*' dan pagar '#'. Tombol-tombol yang ditekan merupakan angka-angka bernilai 1 sampai dengan 9. Dari hasil pengujian tersebut, diketahui bahwa saat penekanan angka 2 dan 3 dilakukan, *seven segment* tidak menampilkan angka 2 dan 3 yang sempurna. Saat angka 2 ditekan, nilai yang muncul pada *seven segment* adalah angka 0. Sedangkan saat angka 3 ditekan, nilai yang muncul adalah angka 1.

Pengecekan *port* 1 dengan multimeter telah sesuai dengan hasil yang diharapkan, yaitu nilai-nilai pada pin-pin *port* 1 telah sesuai dengan nilai-nilai hasil penekanan tombol *keypad*. Dapat disimpulkan bahwa kesalahan tersebut bukan terjadi karena performa *port* 1 yang kurang baik, namun disebabkan alasan lain. Untuk menganalisa lebih jauh mengenai hal tersebut, dilakukan pada bagian analisa rangkaian *keypad*.

c. Port 2

Port 2 sebagai *port* yang berfungsi sebagai *port* yang dihubungkan dengan *keypad*, diuji dengan cara yang sama dengan *port* 1, yaitu dengan menggunakan multimeter. Hasil pengujian menunjukkan hasil yang baik, yaitu nilai-nilai pada pin-pin *port* 2 telah sesuai dengan nilai hasil penekanan tombol *keypad*. Sehingga dapat dikatakan performa *port* 2 pada rangkaian ini cukup baik.

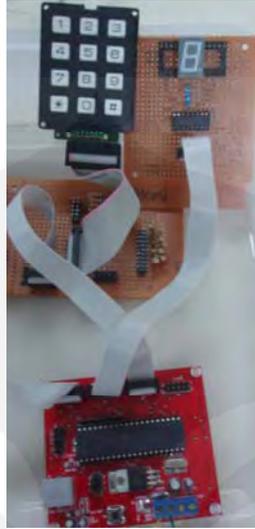
d. Chip Mikrokontroler AT89S51

Pengujian dan analisa untuk *chip* mikrokontroler system minimum program *keypad* ini sama dengan pengujian dan analisa yang dilakukan pada *chip* mikrokontroler system minimum program utama. Hal ini dikarenakan kedua *chip* ini memiliki tipe, spesifikasi, dan fungsi yang sama.

Dari hasil pengujian yang dilakukan, diketahui bahwa performa *chip* mikrokontroler ini cukup baik.

4.1.2 Sub-sistem Rangkaian Keypad

Rangkaian *keypad* dalam kesatuan rangkaian simulator pensinyalan komunikasi seluler bertindak sebagai input yang menandakan proses panggilan keluar telah dimulai. *Keypad* yang digunakan adalah jenis DTMF *keypad* ukuran tiga kolom dan empat baris. Performa dari rangkaian *keypad* ini dapat diketahui dari output yang dihasilkannya yaitu nilai-nilai yang merepresentasikan nomor-nomor *mobile station* yang menjadi tujuan panggilan. Berikut ini adalah gambar sub system rangkaian *keypad*.



Gambar 4.5 sub system rangkaian keypad

Pada Gambar 4.5 diatas, terlihat gambar sub system rangkaian *keypad* yang merupakan bagian dari keseluruhan rangkaian simulator pensinyalan.

Proses penyambungan *keypad* dengan sistem minimum AT89S51 tidak terlalu sulit karena hanya menggunakan *header* yang dipasang pada PCB *board* berlubang. *Keypad* ini tidak bisa langsung dihubungkan dengan *port 2* sistem minimum karena pin yang dimiliki *keypad*, yaitu sebanyak tujuh pin, tidak sesuai dengan pin *port 2*, yang terdiri dari sepuluh pin, sehingga perlu dibuat semacam perantara untuk menghubungkan tujuh pin *keypad* dengan sepuluh pin *port 2*. Perantara yang digunakan adalah dua buah *header* masing-masing dengan tujuh pin, yang dihubungkan ke *keypad*, dan sepuluh pin, yang dihubungkan ke *port 2*. Kedua *header* ini dipasang pada PCB *board* berlubang dan pin-pin yang bersesuaian disambungkan dengan kabel setelah sebelumnya disolder dengan timah. Mekanisme penyambungan pin *header keypad* dengan pin *header port 2* dapat dilihat kembali pada bab 3 bagian sub bab 3.1.2.2

Setelah *keypad* tersambung dengan *port 2* pada sistem minimum AT89S51, maka pengujian performa *keypad* dapat dilakukan. *Keypad* yang digunakan ini dimaksudkan sebagai pengganti tombol-tombol *keypad* yang ada pada ponsel. Karena pada skripsi ini proses panggilan yang dibahas adalah panggilan keluar, maka tombol *keypad* digunakan sebagai input untuk memanggil nomor ponsel tujuan. Kekurangan yang ada pada *keypad* ini adalah tidak adanya tombol sebagai fungsi *dial* sebagaimana yang ada pada ponsel secara umum.

Namun tombol fungsi *dial* tersebut digantikan dengan tombol bintang (*asterisk* “*”). Kekurangan ini diharapkan dapat ditutupi pada penyempurnaan sistem berikutnya.

Selain itu, kesulitan yang dihadapi saat pembuatan sub-sistem rangkaian *keypad* ini terjadi saat pembuatan program untuk menggabungkan program penekanan salah satu tombol *keypad* dengan program pensinyalan utama. Sebagaimana telah dijelaskan pada bab 2 yaitu dasar teori, bahasa pemrograman yang digunakan pada skripsi ini adalah *assembly*. Dalam bahasa *assembly* tidak dikenal adanya pemrograman modular, yang artinya program yang ukurannya besar dapat dipecah-pecah menjadi program yang lebih kecil sehingga dapat dipanggil bila sewaktu-waktu dibutuhkan. Instruksi yang biasa digunakan untuk pemrograman modular adalah *include* yang diteruskan dengan nama *file* yang akan dipanggil. Kesulitan yang dihadapi adalah menggabungkan program penekanan *keypad* dengan program pensinyalan utama kedalam satu program. Beberapa kali ditemui kegagalan saat pembuatan program ini. Kegagalan yang ada biasanya ditandai dengan tidak munculnya nilai-nilai bit pensinyalan yang seharusnya muncul bila tombol *dial* (dalam hal ini tombol bintang) ditekan, karena penekanan tombol ini menandai proses pensinyalan yang seharusnya sudah dimulai. Namun, setelah program tersebut beberapa kali diubah dan diperbaiki, didapatkan program dengan hasil sesuai dengan yang diinginkan.

Kelemahan yang ada pada program rangkaian *keypad* ini adalah tidak adanya pembatasan jumlah nomor yang dapat dimasukkan sebagai input yang menandakan proses panggilan terjadi. Sebagaimana umumnya, nomor-nomor ponsel yang ada saat ini di Indonesia memiliki aturan penomoran khusus yang disesuaikan dengan aturan penomoran internasional. Namun, dalam skripsi ini hal tersebut tidak dapat diterapkan karena literature aturan khusus penomoran tersebut tidak didapatkan seluruhnya. Selain itu, walaupun aturan khusus penomoran tersebut telah diketahui, untuk dapat diterapkan pada skripsi ini tidak memungkinkan karena akan lebih banyak sub sistem komponen pensinyalan yang terlibat seperti BTS, HLR, VLR, AuC, EIR, dan lain lain.

Kekurangan yang cukup terlihat pada rangkaian *keypad* ini adalah saat dilakukan pengujian dengan menekan semua tombol-tombol angka pada *keypad*.

Seperti telah disinggung pada analisa *port 1* sistem minimum program *keypad*, ditemui ketidakcocokan nilai yang ditampilkan pada *seven segment* dengan hasil penekanan angka 2 dan 3. Setelah dilakukan pengecekan terhadap rangkaian *seven segment* dan rangkaian *keypad* yang disolder dan dihubungkan dengan kabel tembaga serabut, tidak ditemukan adanya kesalahan. Begitu juga saat konektor yang menghubungkan rangkaian *seven segment* dengan sistem minimum diperiksa, tidak ditemukan adanya kesalahan. Kesalahan baru ditemukan saat konektor yang menghubungkan *keypad* dengan *header* pada PCB berlubang diperiksa. Ternyata konektor tersebut tidak memiliki performa yang baik sehingga sering mengalami ketidaksesuaian dengan *header*. Solusi yang dilakukan untuk mengatasi hal ini adalah dengan mengganti konektor tersebut dan selalu melakukan pengecekan saat rangkaian akan dijalankan.

Setelah pengujian performa rangkaian *keypad* dilakukan, secara umum didapatkan hasil yang cukup baik walaupun masih terdapat beberapa kekurangan seperti yang telah disebutkan di atas. Kekurangan-kekurangan tersebut diharapkan dapat disempurnakan pada pengembangan rangkaian simulator sistem pensinyalan komunikasi seluler ini selanjutnya.

4.1.3 Sub-sistem Rangkaian Seven Segment

Rangkaian *seven segment* pada kesatuan rangkaian simulator sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim memiliki fungsi untuk menampilkan nilai bit-bit penting yang ada selama proses pensinyalan berlangsung. Selain itu juga untuk menampilkan nilai hasil penekanan salah satu tombol *keypad*. Berikut ini adalah bit-bit penting yang ditampilkan rangkaian *seven segment* selama proses pensinyalan berlangsung.

Tabel 4.1. Tabel bit-bit *message type* dari *message flow* pensinyalan

NO	<i>Message Type</i>	<i>Binary Number</i>	<i>Decimal Number</i>
1	<i>Connection Request</i>	0000 0001	1
2	<i>Location Updating Request</i>	0000 1000	8
3	<i>Connection Confirm</i>	0000 0011	3
4	<i>Identity Request</i>	0001 1000	24

5	<i>IMSI Allocation</i>	0000 1111	15
6	<i>TMSI Allocation</i>	0100 1011	75
7	<i>LAI Request</i>	0001 1001	25
8	<i>LAI Allocation</i>	0000 0100	4
9	<i>Paging (Cari No Tujuan)</i>	1000 0001	129
10	<i>Connection Request Rejected</i>	0000 0010	2
11	<i>Service Request</i>	0010 0100	36
12	<i>Calling Confirmation</i>	1101 1011	219

Dari tabel 4.1 diatas, dapat dilihat terdapat 12 *message type* yang ada selama proses pensinyalan komunikasi seluler bagian *subscriber* pengirim berlangsung. Nilai-nilai inilah yang ditampilkan pada rangkaian *seven segment*. Pada tabel 4.1 tersebut, nilai bit-bit dari *message type* berupa nilai biner. Namun, yang ditampilkan pada rangkaian *seven segment* adalah nilai desimalnya. Berikut ini adalah gambar sub system rangkaian *seven segment*.



Gambar 4.6 sub system rangkaian seven segment

Gambar 4.6 menunjukkan gambar sub system rangkaian *seven segment*. Terlihat bahwa *port 0* (kabel paling kiri) dihubungkan ke satu buah *seven segment* dan *port 1* (kabel paling kanan) dihubungkan ke dua buah *seven segment*.

Nilai decimal terbesar yang ditampilkan bernilai ratusan. Maka digunakan tiga buah *seven segment* untuk dapat menampilkan semua nilai-nilai tersebut. Mekanisme penampilan nilai-nilai tersebut adalah sebagai berikut.

1. Rangkaian *seven segment* dihubungkan dengan dua *port* pada sistem minimum AT89S51, yaitu *port 0* dan *port 1*
2. *Port 0* menampilkan angka decimal bernilai ratusan sedangkan *port 1* menampilkan angka decimal bernilai puluhan dan satuan

Mekanisme diatas dibuat karena angka decimal terbesar yang ditampilkan, dalam hal ini bernilai ratusan, dapat direpresentasikan dengan tiga digit, yang masing-masing digit terdiri dari empat bit. Sementara satu port terdiri dari delapan bit, yang berarti dua digit, sehingga dapat menampilkan dua digit sekaligus. Oleh karena itulah, pada rangkaian *seven segment* ini *port 1* menampilkan dua digit sekaligus, yaitu digit yang merepresentasikan angka decimal bernilai puluhan dan digit yang merepresentasikan angka desial bernilai satuan. Sedangkan *port 0* hanya menampilkan satu digit saja, yaitu yang merepresentasikan angka decimal bernilai ratusan. Alasan penggunaan dua digit dalam satu *port* adalah untuk efektifitas karena *port* yang tersisa dapat digunakan untuk rangkaian lain.

Performa dari rangkaian *seven segment* diuji melalui seberapa tepat angka yang ditampilkan. Pengujian ini meliputi pengecekan tiga buah *seven segment*, tiga buah BCD to *seven segment decoder* 74LS47N, dan rangkaian kabel tembaga serabut yang menghubungkan *decoder*, *seven segment*, dan *header*.

Dari hasil pengujian yang dilakukan, sempat ditemui beberapa kegagalan yaitu tampilan angka-angka pada *seven segment* yang tidak sempurna, yaitu pada *seven segment* dengan angka decimal bernilai puluhan. Setelah dilakukan pengecekan ulang, diketahui bahwa ternyata ke tidak-akuratan tersebut terjadi karena pada sambungan kabel tembaga serabut ada bagian yang saling menempel, sehingga terjadi hubungan singkat (*short circuit*). Selain itu, kegagalan tersebut juga disebabkan rusaknya *decoder* yang terhubung dengan *seven segment* yang menampilkan angka decimal bernilai puluhan. Namun, setelah sambungan kabel tembaga serabut diperbaiki dan *decoder* yang rusak diganti, hasil tampilan angka-angka decimal pada ketiga *seven segment* menjadi sesuai dengan yang diinginkan. Sehingga dapat disimpulkan bahwa performa rangkaian *seven segment* sebagai bagian dari kesatuan rangkaian simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim cukup baik.

4.2 HASIL UJI COBA SISTEM KESELURUHAN

Setelah dilakukan pengujian dan analisa untuk masing-masing sub system rangkaian simulator pensinyalan komunikasi seluler bagian *subscriber* pengirim, berikut ini akan dijelaskan mengenai tahap-tahap pengujian dan analisa untuk rangkaian simulator secara menyeluruh. Karena pada skripsi ini rancang bangun dilakukan dengan *software* dan *hardware*, maka pengujian dan analisa keseluruhan dibagi dalam dua hal tersebut

4.2.2 Uji Coba Software 8051IDE

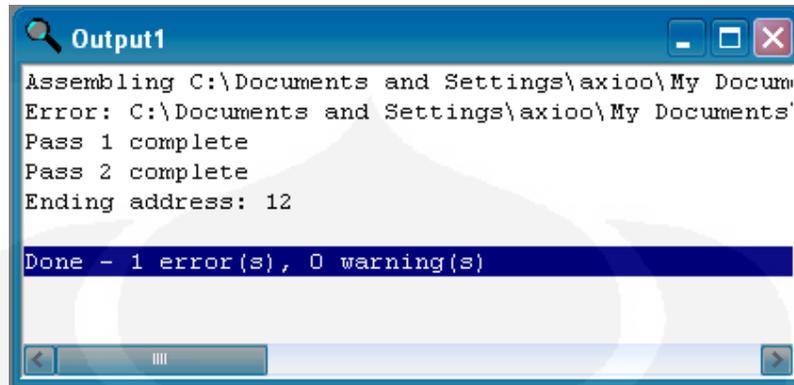
Pengujian pada *software* 8051IDE dilakukan untuk mengetahui jalannya simulasi dari program yang telah dibuat. Selain itu, dengan menggunakan *software* ini, dapat diketahui *error* atau kesalahan yang terjadi selama program dibuat. Bagian-bagian *software* yang diuji dan dianalisa adalah :

1. Output, bagian ini menunjukkan apakah program yang dibuat memiliki kesalahan atau tidak.
2. Register, bagian ini menunjukkan nilai-nilai dari register yang digunakan dalam program.
3. Internal memori (RAM), bagian ini menunjukkan nilai-nilai yang disimpan pada internal memori.
4. *Port*, bagian ini menunjukkan nilai-nilai yang akan muncul pada *port-port* yang digunakan.

Untuk mengetahui lebih jelas mengenai pengujian dan analisa bagian-bagian tersebut, berikut ini penjelasannya.

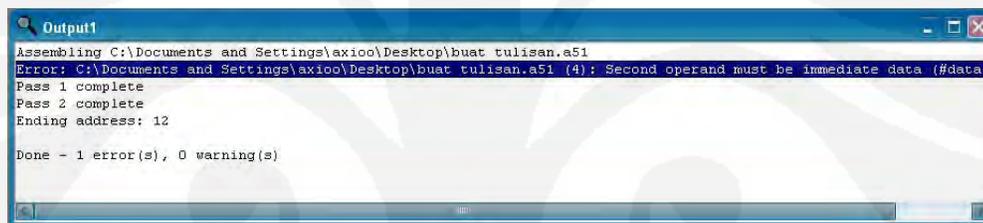
a. Output

Pada *software* 8051IDE, output memiliki fungsi yang sangat penting. Karena bagian ini menunjukkan *error* yang terjadi selama pembuatan program. Selain itu, bagian output juga dapat memberitahukan letak kesalahan pada listing program sekaligus jenis kesalahan tersebut. Gambar 4.7 menunjukkan tampilan output yang muncul bila terjadi kesalahan.



Gambar 4.7 Tampilan output saat terjadi kesalahan

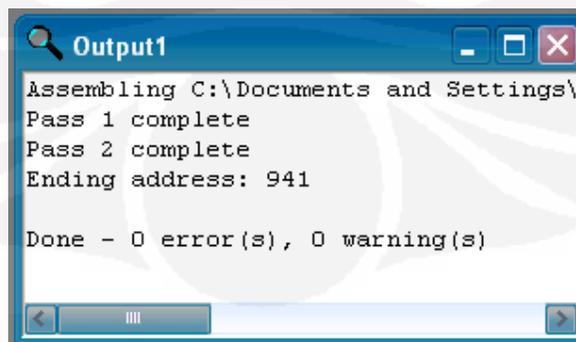
Pada Gambar 4.7 tersebut diketahui adanya satu kesalahan pada program yang telah dibuat. Kesalahan tersebut dapat diketahui letak dan jenisnya dengan men-*double click* baris yang diawali dengan kata *error*. Berikut ini tampilan yang menunjukkan letak dan jenis kesalahan yang terjadi.



Gambar 4.8 Tampilan letak dan jenis kesalahan program

Pada Gambar 4.8 tersebut terlihat bahwa baris yang diawali dengan kata *error* menunjukkan letak kesalahan dan jenis dari kesalahan tersebut. Dari Gambar 4.8 tersebut dapat diketahui bahwa kesalahan terjadi pada baris keempat, yang ditandai dengan angka 4 di dalam kurung, dan jenis kesalahannya adalah data pada operand kedua harus berupa data langsung.

Program pensinyalan utama dan program *keypad* telah disimulasikan pada *software* 8051IDE dan menghasilkan tampilan sebagai berikut.



Gambar 4.9 Tampilan letak dan jenis kesalahan program

Gambar 4.9 menunjukkan bahwa program pensinyalan utama dan program *keypad* telah disimulasikan pada *software* 8051IDE dan tidak terdapat kesalahan pada program tersebut.

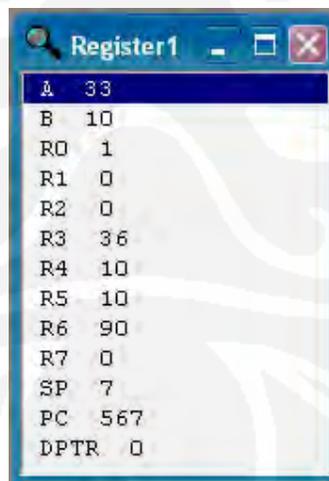
Dari hasil pengujian bagian output *software* 8051IDE, dapat dikatakan bahwa program telah berhasil dengan baik.

b. Register

Pada *software* 8051IDE, register-register yang digunakan dalam program pensinyalan dan program *keypad* adalah sebagai berikut:

- Register 0 (R0) : *delay*
- Register 1 (R1) : *delay*
- Register 2 (R2) : *delay*
- Register 3 (R3) : menyimpan sementara
- Register 4 (R4) : baca
- Register 5 (R5) : kirim
- Register 6 (R6) : menyimpan sementara

Sedangkan register yang tidak digunakan adalah register 7 (R7). Saat disimulasikan, nilai-nilai register ini akan berubah-ubah sesuai dengan nilai yang dimasukkan ke dalamnya. Berikut ini adalah tampilan register dari bagian program yang disimulasikan.



Gambar 4.10 Tampilan register

Dari Gambar 4.10 dapat dilihat nilai-nilai yang muncul untuk masing-masing register. Setelah dilakukan pengujian dengan menjalankan keseluruhan

program yang telah dibuat, dapat diketahui bahwa nilai-nilai yang muncul pada masing-masing register telah sesuai dengan yang diharapkan. Hal tersebut berarti jalannya simulasi pada bagian register telah berhasil dengan baik.

c. Internal memori (RAM)

Bagian internal memori (IRAM) pada *software* 8051IDE berfungsi untuk menyimpan nilai-nilai penting yang ada dalam program, baik yang ditampilkan pada *seven segment*, maupun yang tidak ditampilkan. Nilai-nilai yang tidak ditampilkan pada *seven segment* ditunjukkan pada tabel berikut ini.

Tabel 4.2. Nilai yang tidak ditampilkan

Jenis	Alamat Memori	Nilai
IMSI	10h – 1eh	510 10 91 22112199
TMSI	40h – 4eh	123581321345589
IMEI	50h – 5eh	987654321123456
LAI	70h – 74h	51010
Blok Saluran Informasi	60h – 69h	Tergantung nilai sentral

Nilai-nilai tersebut tidak semuanya merupakan nilai yang valid. Karena beberapa nilai tidak ada dalam data yang didapat dari salah satu vendor telekomunikasi di Indonesia. Nilai-nilai yang valid adalah nilai IMSI, LAI, dan blok saluran informasi. Sedangkan nilai yang diasumsikan adalah TMSI dan IMEI.

Untuk nilai-nilai yang ditampilkan pada *seven segment*, dapat dilihat pada tabel dibawah ini.

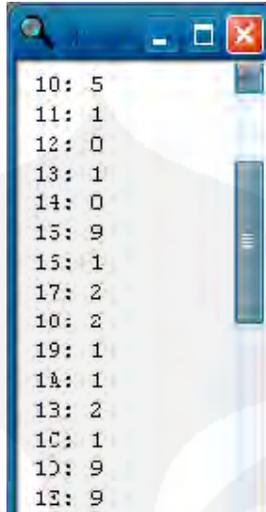
Tabel 4.3. Nilai yang ditampilkan

No	Message Type	Label Program	Nilai
1	IMSI Allocation	alokasi_imsi	15
2	IMSI Allocation Approval	persetujuan_alokasi_imsi	240
3	End of TMSI	akhir_tmsi	51
4	TMSI Allocation	alokasi_tmsi_selesai	75

5	<i>Identity Number</i>	nomor_imei	231
6	<i>Identity Request</i>	minta_identitas	24
7	<i>LAI Check Request</i>	minta_cek_lai	7
8	<i>LAI Request</i>	pengecekan_lai	25
9	<i>Location Updating Request</i>	minta_update_lokasi	8
10	<i>LAI Allocation</i>	alokasi_lai	4
11	<i>Waiting</i>	menunggu_notujuan_dicari	254
12	<i>Paging</i>	paging_carinotuju	129
13	<i>Connection Request</i>	minta_kanal	1
14	<i>Connection Confirm</i>	persetujuan_bangun_kanal	3
15	<i>Service Request Approval</i>	persetujuan_minta_layanan	36
16	<i>Calling Confirmation</i>	konfirmasi_panggilan	9
17	<i>Destination Available</i>	notujuan_dpt_dihubungi	90
18	<i>Connected</i>	terhubung	33
19	<i>Communication</i>	komunikasi	86

Nilai-nilai pada tabel 4.3 diatas merupakan nilai valid yang didapatkan dari salah satu vendor telekomunikasi di Indonesia. Nilai-nilai tersebut merupakan nilai-nilai yang terdapat pada sentral dan *mobile station*.

Pengujian dilaksanakan dengan mensimulasikan program pensinyalan pada *software* 8051IDE. Dari hasil pengujian yang dilakukan, didapatkan tampilan *window* untuk internal memori (IRAM) sebagai berikut.



Gambar 4.11 Tampilan internal memori

Gambar 4.11 diatas merupakan tampilan dari bagian hasil simulasi program pensinyalan pada *software* 8051IDE. Pada gambar tersebut dapat dilihat bahwa nilai yang muncul pada alamat memori 10h sampai dengan 1eh bernilai sama dengan nomor IMSI yang diinginkan. Sehingga dapat dikatakan bahwa performa dari internal memori pada *software* 8051IDE sudah cukup baik.

d. Port

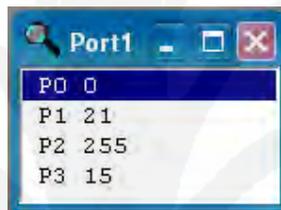
Langkah yang dilakukan pada pengujian bagian *port* dari *software* 8051IDE sama dengan langkah yang dilakukan pada pengujian output, register, dan internal memori. Setelah program disimulasikan dengan *software* 8051IDE, maka akan dilihat nilai-nilai yang muncul pada *port-port* yang digunakan dalam program. *Port-port* yang digunakan pada program pensinyalan utama berbeda dengan yang digunakan pada program *keypad*. Untuk lebih jelasnya, dapat dilihat pada tabel berikut ini.

Tabel 4.4. perbedaan fungsi *port* pada rangkaian *keypad* dengan rangkaian utama

Port	Fungsi	
	Rangkaian Keypad	Rangkaian Utama
0	Sarana penghubung dengan rangkaian utama (dihubungkan dengan port 2 rangkaian utama)	Menampilkan bit penting pensinyalan yang bernilai decimal ratusan

1	Menampilkan nilai hasil penekanan tombol <i>keypad</i> pada <i>seven segment</i>	Menampilkan bit penting pensinyalan yang bernilai decimal puluhan dan satuan
2	Menghubungkan system minimum dengan <i>keypad</i>	Sarana penghubung dengan rangkaian <i>keypad</i> (dihubungkan dengan <i>port 0</i> rangkaian <i>keypad</i>)
3	Tidak digunakan	Sebagai sentral (<i>Mobile Switching Center</i>)

Dari hasil pengujian yang dilakukan pada rangkaian utama, didapatkan hasil tampilan *window* bagian *port* sebagai berikut.



Gambar 4.12 Tampilan Port rangkaian utama

Pada Gambar 4.12 diatas, terlihat bahwa nilai pada *port 0* sampai dengan *port 3* sesuai dengan nilai yang diinginkan yaitu nilai untuk *message type connected* dengan label program terhubung. Perlu diperhatikan bahwa nilai pada *port 0* dan *port 1* merupakan nilai yang ditampilkan pada *seven segment*. Namun nilai yang tertera pada Gambar 4.12 diatas merupakan nilai yang belum melewati prosedur *BCD to seven segment*. Dari hasil pengujian diatas, dapat disimpulkan bahwa performa *port* sebagai bagian *software 8051IDE* berjalan baik.

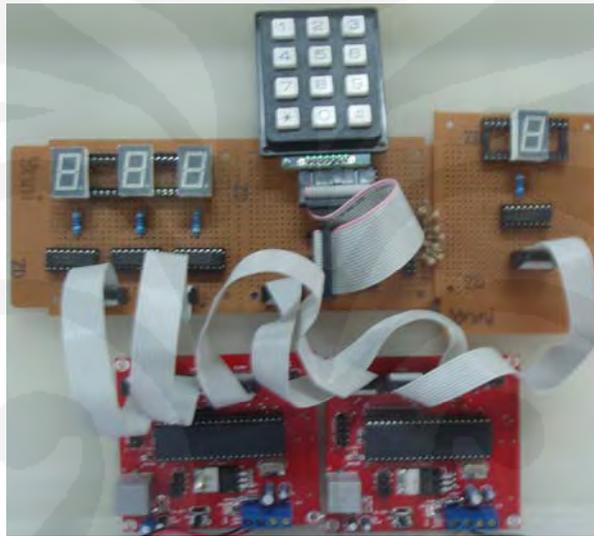
4.2.3 Uji Coba Hardware Rangkaian Simulator Sistem Pensinyalan

Setelah analisa *software 8051IDE* dilakukan, maka analisa selanjutnya adalah hasil uji coba *hardware* rangkaian simulator system pensinyalan komunikasi seluler. Seperti telah dijelaskan pada bab 3 yaitu bab yang menjelaskan rancang bangun dari rangkaian simulator system pensinyalan komunikasi seluler bagian *subscriber* pengirim, rangkaian simulator ini terdiri dari beberapa sub system yaitu, sub system rangkaian system minimum

mikrokontroler AT89S51, sub system rangkaian *seven segment*, dan sub system rangkaian *keypad*.

Pengujian *hardware* rangkaian simulator system pensinyalan komunikasi seluler bagian *subscriber* pengirim dilakukan setelah program pensinyalan dan program *keypad* di-*burn* ke dalam masing-masing mikrokontroler. Tahap-tahap yang dilakukan saat proses *burn* telah dijelaskan pada sub bab 3.1.2.4

Berikut ini adalah gambar rangkaian simulator pensinyalan keseluruhan yang belum melewati proses *burning*.



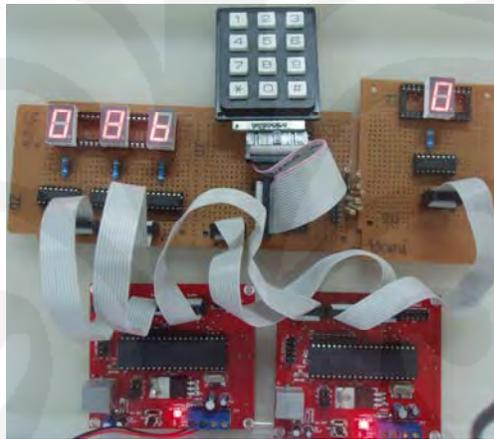
Gambar 4.13 Rangkaian simulator keseluruhan sebelum di-*burn*

Pada Gambar 4.13 diatas, diperlihatkan rangkaian simulator keseluruhan beserta dengan sub systemnya. Tampak pada gambar, system minimum yang berada disebelah kiri terhubung dengan rangkaian *seven segment* yang digunakan untuk menampilkan nilai-nilai dari bit-bit penting pensinyalan. Sedangkan system minimum yang berada di sebelah kanan dihubungkan dengan rangkaian *keypad* yang juga memiliki sebuah *seven segment* untuk menampilkan nilai hasil penekanan tombol *keypad*. Perlu diingat bahwa rangkaian system minimum di sebelah kiri yang terhubung dengan rangkaian *seven segment* merupakan rangkaian pensinyalan utama yang dianalogikan sebagai *mobile station* (MS) dan *mobile switching center* (MSC/sentral). Sedangkan rangkaian system minimum di sebelah kanan yang terhubung dengan rangkaian *keypad* beserta satu buah *seven*

segment merupakan rangkaian kesatuan *keypad* yang dianalogikan sebagai *keypad* pada ponsel sebagai input untuk memasukkan nomor ponsel tujuan.

Terlihat bahwa pada Gambar 4.13, rangkaian pensinyalan utama terhubung dengan rangkaian kesatuan *keypad* melalui konektor antar *port*. *Port* yang dihubungkan adalah *port 2* pada rangkaian pensinyalan utama dengan *port 0* pada rangkaian kesatuan *keypad*.

Saat pengujian *hardware* rangkaian simulator pensinyalan dilakukan, program pensinyalan utama dan program *keypad* di-*burn* kedalam masing-masing rangkaian sub system. Berikut ini adalah gambar yang menunjukkan *hardware* rangkaian simulator pensinyalan setelah di-*burn*.



Gambar 4.14 Rangkaian simulator keseluruhan setelah di-*burn*

Pada Gambar 4.14 diatas, terlihat bahwa nilai-nilai yang diinginkan telah muncul pada rangkaian *seven segment*. Nilai yang tertera pada rangkaian *seven segment* Gambar 4.14 adalah nilai terakhir bit pensinyalan. Angka 86 tersebut merupakan nilai dari *message type communication* dengan label program komunikasi. Sementara nilai yang tertera pada *seven segment* yang terhubung dengan rangkaian *keypad* adalah angka nol. Hal ini merupakan salah satu bentuk kelemahan dari rangkaian simulator pensinyalan ini karena *seven segment* tersebut tidak dapat menampilkan symbol bintang '*'. Solusi yang diberikan untuk permasalahan ini adalah dalam penyempurnaan rangkaian simulator pensinyalan selanjutnya sebaiknya komponen yang digunakan untuk menampilkan nilai-nilai adalah layar LCD dan bukan *seven segment*, karena layar LCD dapat deprogram untuk menampilkan symbol-simbol tertentu seperti tanda bintang '*'.

Secara umum jalannya simulasi bit-bit pensinyalan pada rangkaian simulator ini cukup baik. Hal ini bisa dilihat dari beberapa parameter sebagai berikut:

1. Semua bit-bit pensinyalan penting dapat ditampilkan dengan lengkap dan baik.
2. Interaksi antara system minimum rangkaian *seven segment* dengan system minimum rangkaian kesatuan *keypad* berjalan dengan baik. Hal ini bisa dibuktikan dengan tidak adanya *delay* antara saat tombol bintang telah ditekan dengan tampilnya urutan bit-bit penting pensinyalan.
3. *Keypad* dapat berfungsi sebagai input nomor ponsel tujuan sebagaimana ponsel pada umumnya.

Solusi yang disarankan untuk mengatasi permasalahan pada batasan masalah adalah sebagai berikut:

1. Mencari aturan penomoran ponsel yang digunakan saat ini di Indonesia sehingga dalam penyempurnaan rangkaian simulator pensinyalan tersebut, nomor ponsel tujuan yang digunakan sebagai input memiliki batasan yang jelas.
2. Menggunakan komponen layar LCD sebagai alat untuk menampilkan nilai-nilai penting bit pensinyalan dan hasil penekanan tombol *keypad* sehingga permasalahan tidak dapat ditampilkannya symbol tertentu dapat teratasi.
3. Solusi lain untuk mengatasi permasalahan fungsi *dial* tersebut adalah menggunakan *push button* yang dianalogikan sebagai tombol 'yes' atau 'no' pada ponsel sesungguhnya.
4. Memisahkan program dan rangkaian dari masing-masing komponen pensinyalan seperti *mobile station* dan *mobile switching center* sehingga alur penerimaan dan respon antara keduanya dapat terlihat jelas.
5. Menambahkan komponen pensinyalan lain seperti BTS, BSC, HLR, dan VLR dalam program dan rangkaian terpisah sehingga rangkaian simulator system pensinyalan dapat merepresentasikan system pensinyalan yang sesungguhnya.

4.3 ANALISA KONDISI PROSES PENSINYALAN PADA RANGKAIAN SIMULATOR SISTEM PENSINYALAN KOMUNIKASI SELULER BAGIAN SUBSCRIBER PENGIRIM

Dalam rancang bangun system pensinyalan komunikasi seluler bagian *subscriber* pengirim ini terdapat beberapa kondisi yang harus dilaksanakan supaya proses pensinyalan dapat berjalan dengan lancar dan sesuai dengan system sebenarnya. Kondisi-kondisi tersebut adalah sebagai berikut :

- a. Kondisi *idle*, yang terbagi lagi menjadi tiga tahap, yaitu :
 1. IMSI *attach*
 2. Identifikasi IMEI
 3. Perbaruan lokasi
- b. Kondisi panggilan, dalam hal ini adalah panggilan keluar karena rancang bangun dilakukan pada bagian *subscriber* pengirim.

Berikut ini akan dianalisa satu persatu masing-masing kondisi yang terjadi pada rancang bangun system pensinyalan komunikasi seluler yang dibuat.

4.3.1 Kondisi IMSI Attach

Pada kondisi ini, MS sebagai *subscriber* pengirim akan mengirimkan nomor IMSI ke sentral dan akan dilanjutkan dengan pengiriman nomor TMSI dari sentral ke MS. Dalam program yang dibuat dan dimasukkan ke dalam mikrokontroller, kedua tindakan tersebut telah diikutsertakan. Setelah pengujian dilakukan dengan mensimulasikan program pensinyalan tersebut, hasil yang didapatkan telah sesuai dengan hasil yang diharapkan. Namun masih terdapat beberapa kekurangan dalam kondisi ini, diantaranya:

1. Nomor TMSI yang tidak sesuai dengan nomor TMSI yang digunakan pada system pensinyalan di Indonesia sehingga dalam program yang dibuat, nomor TMSI ini diasumsikan.
2. Ketidaksesuaian kondisi ini dengan kondisi IMSI *attach* yang sesungguhnya.

Pada poin kedua dalam hal-hal yang menjadi kekurangan kondisi IMSI *attach* rancang bangun ini adalah ketidaksesuaian kondisi ini dengan kondisi IMSI

attach yang sesungguhnya. Ketidaksesuaian ini dapat dilihat dari tiga parameter, yaitu:

1. Frekuensi BTS

Dalam kondisi sesungguhnya, saat kondisi IMSI *attach* MS melakukan pencarian frekuensi BTS terdekat yang dapat melayani MS tersebut dengan baik. Pencarian frekuensi ini penting karena tidak semua BTS memiliki kesesuaian frekuensi dengan MS tertentu.

2. Penerimaan kuat sinyal BTS

Prosedur ini memungkinkan MS mendapat layanan terbaik dalam hal penerimaan kuat sinyal. Dalam kondisi sesungguhnya, setelah MS mencocokkan frekuensi dengan beberapa BTS terdekat, maka MS akan melakukan pemilihan BTS yang memberikan layanan sinyal terbaik. Namun setelah MS menentukan satu BTS yang *servicing* kepada MS tersebut, MS terus memantau BTS-BTS lain (yang sesuai frekuensinya) untuk tindakan *preventif* apabila MS tersebut melakukan mobilitas antar BTS.

3. Penyimpanan nomor IMSI, TMSI, dan LAI

Dalam rancang bangun ini, penyimpanan nomor IMSI sudah mengikuti prosedur yang benar, namun untuk TMSI dan LAI masih belum sesuai dengan kondisi sebenarnya. Nomor TMSI seharusnya baru ada setelah MS mengirimkan nomor IMSI yang disimpan di HLR, yang merupakan bagian dari MSC atau sentral. Namun pada rancang bangun ini nomor TMSI telah tersimpan di memori internal mikrokontroler bersama-sama dengan nomor IMSI. Ketidaksesuaian selanjutnya adalah pada program yang dibuat, nomor LAI baru diakses saat kondisi perbaruan lokasi. Namun dalam kondisi sesungguhnya, nomor LAI seharusnya sudah ada saat MS menentukan BTS mana yang *servicing* kepada MS tersebut.

Dari hasil analisa diatas, dapat disimpulkan bahwa performa program bagian kondisi IMSI *attach* pada rancang bangun ini kurang baik karena masih banyak memiliki kekurangan yang perlu disempurnakan.

4.3.2 Kondisi Identifikasi IMEI

Pada kondisi ini, MS diminta untuk memberikan nomor identitas yang disebut IMEI untuk diteliti keabsahannya. Nomor IMEI ini akan diperiksa oleh HLR yang ada di sentral. Dalam kondisi sesungguhnya, IMEI bisa saja masuk kategori *black list* sehingga tidak dapat digunakan. Namun pada rancang bangun ini, tidak dibuat prosedur pengecekan keabsahan IMEI, sehingga IMEI otomatis dianggap sah. Hal tersebut menjadi kekurangan yang harus diperbaiki pada penyempurnaan rancang bangun berikutnya.

4.3.3 Kondisi Perbaruan Lokasi

Kondisi yang terjadi saat *idle* ini seharusnya dilakukan apabila terjadi perubahan nomor LAI. perubahan tersebut dilakukan setelah adanya pengecekan nomor LAI yang tersimpan pada SIM *card* dengan nomor LAI yang menunjukkan sel yang sedang memberikan layanan pada MS tersebut. Selain itu, apabila ditemui perbedaan nomor LAI tersebut, maka sentral akan mengirimkan nomor TMSI baru sebagai nomor sementara pada MS. Pada rancang bangun ini, prosedur tersebut tidak dapat dilakukan karena keterbatasan memori pada mikrokontroler yang digunakan.

4.3.4 Kondisi Panggilan Keluar

Pada kondisi ini, MS melakukan tindakan untuk memanggil MS lain maupun nomor yang ada pada jaringan PSTN. Pada rancang bangun ini telah dibuat beberapa penyesuaian yang memungkinkan program pensinyalan mendekati keadaan sesungguhnya, diantaranya:

1. Kondisi serah terima *message type* antara sentral dengan MS sudah dapat dilakukan. Contohnya adalah MS meminta kanal pensinyalan untuk membangun komunikasi, maka sentral memberikan persetujuan pembangunan kanal pensinyalan.
2. MS tujuan sudah dibuat dalam dua kondisi, yaitu sibuk dan dapat menerima panggilan.

Namun, selain penyesuaian tersebut, dalam rancang bangun ini juga masih ditemui beberapa kekurangan, diantaranya:

1. Pada program pensinyalan, tidak dibuat adanya prosedur autentikasi. Hal ini disebabkan tidak didapatkannya algoritma autentikasi yang khusus dimiliki perusahaan-perusahaan telekomunikasi yang memiliki system pensinyalan.
2. Input nomor tujuan pada *keypad* masih belum menggunakan aturan penomoran yang umum digunakan di Indonesia.

Kelebihan dan kekurangan yang telah disebutkan diharapkan dapat dijadikan pertimbangan dalam proses penyempurnaan rangkaian simulator system pensinyalan komunikasi seluler di Indonesia.

BAB V

KESIMPULAN

Setelah melakukan rancang bangun dan analisis terhadap kinerja rangkaian simulator sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim yang dibuat dengan menggunakan *software* 8051IDE dan *hardware* mikrokontroler AT89S51, maka diperoleh beberapa kesimpulan, yaitu :

1. Rangkaian simulator sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim yang dibuat telah dapat menerapkan prinsip-prinsip dasar pensinyalan komunikasi seluler di Indonesia.
2. Secara umum, rangkaian simulator sistem pensinyalan komunikasi seluler bagian *subscriber* pengirim yang dibuat sudah dapat merepresentasikan sistem pensinyalan komunikasi seluler yang saat ini digunakan di Indonesia sehingga dapat dijadikan dasar dalam pembuatan sistem pensinyalan komunikasi seluler yang lengkap dengan memasukkan pensinyalan pada *subscriber* penerima.

DAFTAR ACUAN

- [1] Gunawan, Andi, "Sharing Knowledge," Jakarta : PT. INDOSAT, 2007
- [2] Usman, Ule Kurniawan, "Sistem Komunikasi Bergerak," Bandung : Lab SISKOM – STT Telkom
www.stttelkom.ac.id/staf/UKU/Handout%20PT1123-DAS
- [3] _____, "Teknologi Seluler," Diakses 10 Juni 2008
http://www.e-dukasi.net/pengpop/pp_full.php?ppid=208&fname=semua.htm
- [4] _____, "Comon Channel Signalling," Bahan Kuliah Jaringan Telekomunikasi, Bandung : Program Studi Teknik Telekomunikasi JTE – STT TELKOM, 2006
- [5] _____, GSM System, White Paper : Nokia, 2005
- [6] GSM Communication Flow, Wireless Curriculum Development Section,
<http://www.huawei.com>
- [7] Usman, Ule Kurniawan, *Basic Call Setup pada GSM dan GPRS*, Lab SISKOM-STT Telkom
- [8] _____, System Training : GSM Traffic Management, Training Document NOKIA, 2002
- [9] Javier Gozalvez Sempere, *An Overview of the GSM System*, Scotland : University of Strathclyde
- [10] Gunnar Heine, *GSM Network : Protocols, Terminologi, and Implementation* (London : Artech House, 1998)
- [11] Data Sheet Mikrokontroller Atmel 89S51

DAFTAR PUSTAKA

- _____, *GSM System*, White Paper : Nokia, 2005
- GSM Communication Flow, *Wireless Curriculum Development Section*,
www.huawei.com
- Gunawan, Andi, *Sharing Knowledge*, PT INDOSAT, Jakarta, 2007
- _____, *Global System for Mobile Communication*, Diakses 20 Mei 2008 dari
The International Engineering Consortium.
<http://www.iec.org>
- _____, *Teknologi Jaringan GSM*, Diakses 25 Mei 2008 dari WanXp
Homepage (2007)
<http://root.wanxp.net>
- _____, *Arsitektur Mikrokontroler AT89S51*. Diakses 2 Mei 2008, dari
Mytutorialcafe Homepage (2004)
<http://www.mytutorialcafe.com/arsitektur/mikrokontroler/AT89S51/303788.pdf>
- Heine, Gunnar, *GSM Network : Protocols, Terminologi, and Implementation*,
London : Artech House, 1998
- GSM and Personal Communication Handbook. Siegmund M. Redl, Matthias K.
Weber, Malcom W. Oliphant. Artech House. London. 1998
- _____, *System Training : GSM Traffic Management*, Training Document
NOKIA, 2002
- Fauzi, Suherman Rahmad, *Jaringan Telekomunikasi*, Diakses tanggal 10 Juni 2008
(2006)
- Usman, Ule Kurniawan, *Basic Call Setup pada GSM dan GPRS*, Lab SISKOM-
STT Telkom
- Sempere, Javier Gozalvez, *An Overview of the GSM System*, Scotland : University
of Strathclyde
- _____, *Sistem Komunikasi Bergerak Modul 03 : Arsitektur Dasar dan
Fungsi Perangkat Standar Sistem Seluler*, Bandung, 2006

LAMPIRAN

Lampiran 1 Listing Program Prosedur IMSI Attach

```
; Prosedur IMSI attach
    mov 10h,#0101b ;5
    mov 11h,#0001b ;1
    mov 12h,#0000b ;0
    mov 13h,#0001b ;1
    mov 14h,#0000b ;0
    mov 15h,#1001b ;9
    mov 16h,#0001b ;1
    mov 17h,#0010b ;2
    mov 18h,#0010b ;2
    mov 19h,#0001b ;1
    mov 1ah,#0001b ;1
    mov 1bh,#0010b ;2
    mov 1ch,#0001b ;1
    mov 1dh,#1001b ;9
    mov 1eh,#1001b ;9

;mengirim imsi ke sentral

                                mov p3,alokasi_imsi
    mov a,p3
    call keseluruhan
                                acall sentral

putar1:    mov a,p3
    call keseluruhan
    jz putar1
                                cjne a,#64,putar1
                                mov p3,10h
                                mov p3,11h
                                mov p3,12h
                                mov p3,13h
                                mov p3,14h
                                mov p3,15h
                                mov p3,16h
                                mov p3,17h
                                mov p3,18h
                                mov p3,19h
                                mov p3,1ah
                                mov p3,1bh
                                mov p3,1ch
                                mov p3,1dh
                                mov p3,1eh

pemberian_tmsi_ke_ms:
    mov p3,#0001b ;1
    mov 40h,p3
    mov p3,#0010b ;2
    mov 41h,p3
    mov p3,#0011b ;3
    mov 42h,p3
    mov p3,#0101b ;5
    mov 43h,p3
```

```

mov p3,#1000b ;8
mov 44h,p3
mov p3,#0001b ;1
mov 45h,p3
mov p3,#0011b ;3
mov 46h,p3
mov p3,#0010b ;2
mov 47h,p3
mov p3,#0001b ;1
mov 48h,p3
mov p3,#0011b ;3
mov 49h,p3
mov p3,#0100b ;4
mov 4ah,p3
mov p3,#0101b ;5
mov 4bh,p3
mov p3,#0101b ;5
mov 4ch,p3
mov p3,#1000b ;8
mov 4dh,p3
mov p3,#1001b ;9
mov 4eh,p3

clr a

mov p3,akhir_tmsi
mov a,akhir_tmsi
;mov a,p3
call keseluruhan
acall sentral
putar2: mov a,p3
call keseluruhan
jz putar2
cjne a,#117,putar2

```

Lampiran 2 Listing Program Prosedur Identifikasi ME

```

;IDENTIFIKASI IMEI

mov p3,nomor_imei
mov a,p3
call keseluruhan
acall sentral
rply: mov a,p3
call keseluruhan
cjne a,#36,rply
sjmp nyimei

nyimei: mov 50h,#9
mov 51h,#8
mov 52h,#7
mov 53h,#6
mov 54h,#5
mov 55h,#4
mov 56h,#3
mov 57h,#2
mov 58h,#1
mov 59h,#1
mov 5ah,#2
mov 5bh,#3
mov 5ch,#4
mov 5dh,#5
mov 5eh,#6

;mengirim imei ke sentral

```

```

mov p3,50h
mov p3,51h
mov p3,52h
mov p3,53h
mov p3,54h
mov p3,55h
mov p3,56h
mov p3,57h
mov p3,58h
mov p3,59h
mov p3,5ah
mov p3,5bh
mov p3,5ch
mov p3,5dh
mov p3,5eh
clr a

```

Lampiran 3 Listing Program Prosedur Perbaruan Lokasi

```

;PERBARUAN LOKASI

mov p3,minta_cek_lai
mov a,p3
call keseluruhan
acall sentral

cek:   mov a,p3
call keseluruhan
cjne a,#37,cek

mov 70h,#5
mov 71h,#1
mov 72h,#0
mov 73h,#1
mov 74h,#0

mov p3,#4

;PENGECEKAN PERUBAHAN LAI

bit1:  mov a,p3
cjne a,70h,LU

bit2:  mov a,p3
cjne a,71h,LU

bit3:  mov a,p3
cjne a,72h,LU

bit4:  mov a,p3
cjne a,73h,LU

bit5:  mov a,p3
cjne a,74h,LU

LU:    mov p3,minta_update_lokasi
mov a,p3
call keseluruhan
acall sentral
mov a,p3
call keseluruhan
;cjne a,#4,LU

;ALOKASI LAI

```

```

    mov p3,#4
step1:    mov a,p3
          cjne a,#0,store1
          ;jnz store1
          sjmp step3
store1:   mov 70h,a
          mov p3,#3
step2:   mov a,p3
          cjne a,#0,store2
          ;jnz store2
          sjmp step2
store2:   mov 71h,a
          mov p3,#2
step3:   mov a,p3
          cjne a,#0,store3
          ;jnz store3
          sjmp step3
store3:   mov 72h,a
          mov p3,#1
step4:   mov a,p3
          ;jnz store4
          cjne a,#0,store4
          sjmp step4
store4:   mov 73h,a
          mov p3,#1
step5:   mov a,p3
          cjne a,#0,store5
          ;jnz store5
          sjmp step5
store5:   mov 74h,a
          ;sjmp menunggu

```

Lampiran 4 Listing Program Prosedur Panggilan Keluar

```

;memanggil no tujuan
          clr a
menunggu: mov p3,menunggu_notujuan_dicari
          mov a,p3
          call keseluruhan
          acall sentral
carinotuju: mov a,p3
          call keseluruhan
          jz carinotuju
          cjne a,#41,carinotuju
lanjut:   mov p3,minta_kanal
          mov a,p3
          call keseluruhan

up:       acall sentral

;set Up panggilan
kanal:   mov a,p3
          call keseluruhan
          jz kanal
          cjne a,#3,rjct
          sjmp trs
rjct:    cjne a,permintaan_kanal_ditolak,up
          sjmp lanjut
          mov p3,minta_layanan

          mov a,p3
          call keseluruhan

trs:     acall sentral
set:     mov a,p3
          call keseluruhan
          jz set

```

```

                                cjne a,#54,carinotuju
                                mov p3,konfirmasi_panggilan
    mov a,p3
    call keseluruhan

delay:    acall sentral
          mov a,p3
          call keseluruhan
          jz delay
          cjne a,#144,sibuk

sibuk:    mov p3,notujuan_sibuk
          mov a,p3
          call keseluruhan
          ajmp end

hubungan: acall sentral
          mov a,p3
          call keseluruhan
          cjne a,#51,hubungan

          ajmp end

end:      mov p3,komunikasi
          mov a,p3
          call keseluruhan
          sjmp end

```

Lampiran 5 Listing Program Prosedur Delay

```

;delay 6 detik
mov r2,#120          ;20*50ms=1000ms=1s

t1ms:  mov r0,#76      ;high byte of 19,457 (76 * 256 =
19,456)
        mov r1,#01     ;low byte of 19,457 (19,456 + 1 =
19,457)
        mov tmod,#1    ;timer 0 in 16-bit mode
loop:   mov th0,r0      ;move R0 value to TH0
        mov tl0,r1     ;move R1 value to TL0
        setb tr0       ;make timer 0 start counting
        jnb tf0,$      ;wait until tf0 = 1
        clr tf0        ;clear flag tf0
        djnz r2,loop   ;wait until R2 = 0

```

Lampiran 6 Listing Program Sentral

```

sentral:  inc r5
          mov b,r5
          mov a,r4

;Prosedur Kirim
          cjne a,b,alamat1
          ret
alamat1:  mov r3,blok1
          cjne r3,#0,alamat2
          mov r6,p3
          mov blok1,r6
          acall baca
          ret
alamat2:  mov r3,blok2
          cjne r3,#0,alamat3
          mov r6,p3
          mov blok2,r6
          acall baca
          ret
alamat3:  mov r3,blok3

```

```

        cjne r3,#0,alamat4
        mov r6,p3
        mov blok3,r6
        acall baca
        ret
alamat4:  mov r3,blok4
        cjne r3,#0,alamat5
        mov r6,p3
        mov blok4,r6
        acall baca
        ret
alamat5:  mov r3,blok5
        cjne r3,#0,alamat6
        mov r6,p3
        mov blok5,r6
        acall baca
        ret
alamat6:  mov r3,blok6
        cjne r3,#0,alamat7
        mov r6,p3
        mov blok6,r6
        acall baca
        ret
alamat7:  mov r3,blok7
        cjne r3,#0,alamat8
        mov r6,p3
        mov blok7,r6
        acall baca
        ret
alamat8:  mov r3,blok8
        cjne r3,#0,alamat9
        mov r6,p3
        mov blok8,r6
        acall baca
        ret
alamat9:  mov r3,blok9
        cjne r3,#0,alamat10
        mov r6,p3
        mov blok9,r6
        acall baca
        ret
alamat10: mov r6,p3
        mov blok10,r6
        acall baca10
        ret

;Prosedur baca
baca:    cjne a,b,cek1
        ret
cek1:    cjne r5,#0,cek2
        ret
cek2:    cjne r5,#1,cek3
        ajmp baca1
cek3:    cjne r5,#2,cek4
        ajmp baca2
cek4:    cjne r5,#3,cek5
        ajmp baca3
cek5:    cjne r5,#4,cek6
        ajmp baca4
cek6:    cjne r5,#5,cek7
        ajmp baca5
cek7:    cjne r5,#6,cek8
        ajmp baca6
cek8:    cjne r5,#7,cek9
        ajmp baca7
cek9:    cjne r5,#8,cek10
        ajmp baca8

```

```

cek10:      cjne r5,#9,cek11
            ajmp baca9
cek11:      cjne r5,#10,cek12
            ajmp baca10
cek12:      ret

baca1:      inc r4
            mov p3,persetujuan_alokasi_imsi
            ret
baca2:      inc r4
            mov p3,alokasi_tmsi_selesai
            ret
baca3:      inc r4
            mov p3,minta_identitas
            ret
baca4:      inc r4
            mov p3,pengecekan_lai
            ret
baca5:      inc r4
            mov p3,alokasi_lai
            ret
baca6:      inc r4
            mov p3,paging_carinotuju
            ret
baca7:      inc r4
            mov p3,persetujuan_bangun_kanal
            ret
baca8:      inc r4
            mov p3,persetujuan_minta_layanan
            ret
baca9:      inc r4
            mov p3,notujuan_dpt_dihubungi
            ret
baca10:     inc r4
            mov p3,terhubung
            ret

```

Lampiran 7 Listing Program Deklarasi Variabel dan Deklarasi Alamat

```

awal:      mov a,p2
            cjne a,#00000001b,awal
            sjmp mulai

mulai:
;mobile station poenya
notujuan      equ          00h
Data_Key      equ          P2
alokasi_imsi  equ          20h
akhir_tmsi    equ          21h
nomor_imei    equ          22h
minta_cek_lai equ          23h
minta_update_lokasi equ      24h
minta_kanal   equ          25h
minta_layanan equ          26h
konfirmasi_panggilan equ      27h
menunggu_notujuan_dicari equ    28h

;Sentral (MSC) Poenya

persetujuan_alokasi_imsi equ      29h
alokasi_tmsi_selesai    equ      2ah
minta_identitas         equ      2bh
pengecekan_lai         equ      2ch
alokasi_lai            equ      2dh
notujuan_sibuk         equ      2eh
paging_carinotuju      equ      2fh
persetujuan_bangun_kanal equ      30h

```

```

peretujuan_minta_layanan      equ      31h
notujuan_dpt_dihubungi       equ      32h
permintaan_kanal_ditolak     equ      33h
terhubung      equ  34h
komunikasi      equ  35h

;Saluran Informasi

blok1      equ      60h
blok2      equ      61h
blok3      equ      62h
blok4      equ      63h
blok5      equ      64h
blok6      equ      65h
blok7      equ      66h
blok8      equ      67h
blok9      equ      68h
blok10     equ      69h

;memasukkan bit pada variabel alamat

mov      alokasi_imsi,#00001111b      ;15
mov      akhir_tmsi,#00110011b      ;51
mov      nomor_imei,#11100111b      ;231
mov      minta_cek_lai,#00000111b   ;7
mov      minta_update_lokasi,#00001000b ;8
mov      minta_kanal,#00000001b     ;1
mov      minta_layanan,#00000101b   ;5
mov      konfirmasi_panggilan,#00001001b ;9
mov      menunggu_notujuan_dicari,#11111110b ;254
mov      persetujuan_alokasi_imsi,#11110000b ;240
mov      alokasi_tmsi_selesai,#01001011b ;75
mov      minta_identitas,#00011000b  ;24
mov      pengecekan_lai,#00011001b   ;25
mov      alokasi_lai,#00000100b      ;4
mov      notujuan_sibuk,#10101010b   ;170
mov      paging_carinotuju,#10000001b ;129
mov      persetujuan_bangun_kanal,#00000011b ;3
mov      persetujuan_minta_layanan,#00100100b ;36
mov      notujuan_dpt_dihubungi,#01011010b ;90
mov      permintaan_kanal_ditolak,#00001110b ;14
mov      r0,#0
mov      r1,#0
mov      r2,#0
mov      r3,#0
mov      r4,#0
mov      r5,#0
mov      r6,#0
mov      70h,#0
mov      71h,#0
mov      72h,#0
mov      73h,#0
mov      74h,#0
mov      terhubung,#00100001b      ;33
mov      komunikasi,#86      ;86
mov      blok1,#0
mov      blok2,#0
mov      blok3,#0
mov      blok4,#0
mov      blok5,#0
mov      blok6,#0
mov      blok7,#0
mov      blok8,#0
mov      blok9,#0
mov      blok10,#0

```

Lampiran 8 Listing Program Nilai Pada Sentral

```
;Nilai-nilai pada Sentral

    mov 50h,#9           ;IMEI
    mov 51h,#8
    mov 52h,#7
    mov 53h,#6
    mov 54h,#5
    mov 55h,#4
    mov 56h,#3
    mov 57h,#2
    mov 58h,#1
    mov 59h,#1
    mov 5ah,#2
    mov 5bh,#3
    mov 5ch,#4
    mov 5dh,#5
    mov 5eh,#6

    mov 10h,#0101b      ;5   ;IMSI
    mov 11h,#0001b      ;1
    mov 12h,#0000b      ;0
    mov 13h,#0001b      ;1
    mov 14h,#0000b      ;0
    mov 15h,#1001b      ;9
    mov 16h,#0001b      ;1
    mov 17h,#0010b      ;2
    mov 18h,#0010b      ;2
    mov 19h,#0001b      ;1
    mov 1ah,#0001b      ;1
    mov 1bh,#0010b      ;2
    mov 1ch,#0001b      ;1
    mov 1dh,#1001b      ;9
    mov 1eh,#1001b      ;9

    mov 70h,#5          ;LAI
    mov 71h,#1
    mov 72h,#0
    mov 73h,#1
    mov 74h,#0
```

Lampiran 9 Listing Program Menampilkan Ke Seven Segment

```
;tampilkan nilai keseluruhan ke seven segment
```

```
keseluruhan:
```

```
    mov b,#10
    div ab
    mov r0,b
    mov b,#10
    div ab
    mov r1,a
    mov a,#16
    mul ab
    mov b,r0
    add a,b

    mov p0,r1
    mov pl,a
    ;mov r0,#255
    ;mov r1,#255
    Ret
```

Lampiran 10 Listing Program Keypad

```
notujuan equ 00h                :UDAH OKEH...!!!!!!

Data_Key      EQU P2

Col2  Bit    Data_Key.0
Row1  Bit    Data_Key.1
Col1  Bit    Data_Key.2
Row4  Bit    Data_Key.3
Col3  Bit    Data_Key.4
Row3  Bit    Data_Key.5
Row2  Bit    Data_Key.6

START:
    call Get_Keypad
Enter:
    Acall Sevseg
    ;call Delay
    sjmp START

Get_Keypad:                        ;loopingan ngecek nilai keypad
    Acall cek_kolom                ;Check_Keypad
    mov a,notujuan
    Cjne a,#0AH,sama                ;0FF brarti keypad ga dipencet
    call mulai

sama:  Cjne a,#0FFH,Enter            ;0FF brarti keypad ga dipencet
    Jmp Get_Keypad

Check_Keypad:                       ;bwt ngecek nilai keypad
    ;Mov Data_Key,#0FFH
    Clr Col1
    Clr Col2
    Clr Col3
    ;Acall Delay
    Mov A,Data_Key
    Anl A,#11101010b
    Cjne A,#11101010b,cek_kolom
    Mov A,#0FFH
```

```

Ret

cek_kolom:
    jnb Col1,Column_1_Line_1
    jnb Col2,Column_2_Line_1
    Acall Column_3_Line_1

Check_Column_1:
    Clr Col1
    Setb Col2
    Setb Col3
    Mov A,Data_Key
    Anl A,#1111011b
    Cjne A,#1111011b,Column_1_Line_1
    Jmp Check_Column_2

Column_1_Line_1:
    mov a,data_key
    anl a,#11111011b
    mov data_key,a
    Jb Row1,Column_1_Line_2
    mov notujuan,#1
    ;Jnb Row1,$
    Acall LOnG_Delay
    Mov Data_Key,#0FFH
    Ret

Column_1_Line_2:
    mov a,data_key
    anl a,#11111011b
    mov data_key,a
    Jb Row2,Column_1_Line_3
    mov notujuan,#4
    ;Jnb Row2,$
    Acall LOnG_Delay
    Mov Data_Key,#0FFH
    Ret

Column_1_Line_3:

```

```

mov a,data_key
anl a,#11111011b
mov data_key,a
Jb Row3,Column_1_Line_4
mov notujuan,#7
;Jnb Row3,$
Acall LOng_Delay
Mov Data_Key,#0FFH
Ret

```

Column_1_Line_4:

```

mov a,data_key
anl a,#11111011b
mov data_key,a
Jb Row4,Back
Mov notujuan,#0Ah
;Jnb Row4,$
Acall LOng_Delay
Mov Data_Key,#0FFH
Ret

```

Check_Column_2:

```

Setb Col1
Clr Col2
Setb Col3
;Acall Delay
Mov A,Data_Key
Anl A,#1111110b
Cjne A,#1111110b,Column_2_Line_1
Jmp Check_Column_3

```

Column_2_Line_1:

```

mov a,data_key
anl a,#11111110b
mov data_key,a
Jb Row1,Column_2_Line_2
mov notujuan,#2
;Jnb Row1,$
Acall LOng_Delay

```

```

Mov Data_Key,#0FFH
Ret

Column_2_Line_2:
mov a,data_key
anl a,#11111110b
mov data_key,a
Jb Row2,Column_2_Line_3
Mov notujuan,#5
;Jnb Row2,$
Acall LOng_Delay
Mov Data_Key,#0FFH
Ret

Column_2_Line_3:
mov a,data_key
anl a,#11111110b
mov data_key,a
Jb Row3,Column_2_Line_4
Mov notujuan,#8
;Jnb Row3,$
Acall LOng_Delay
Mov Data_Key,#0FFH
Ret

Back:
Ret

Column_2_Line_4:
mov a,data_key
anl a,#11111110b
mov data_key,a
Jb Row4,Back
Mov notujuan,#0
;Jnb Row4,$
Acall LOng_Delay
Mov Data_Key,#0FFH
Ret

```

```

Check_Column_3:
    Setb Col1
    Setb Col2
    Clr Col3
    ;Acall Delay
    Mov A,Data_Key
    Anl A,#1110111b
    Cjne A,#1110111b,Column_3_Line_1
    Mov A,#0FFH
    Mov Data_Key,#0FFH
    Ret

```

```

Column_3_Line_1:
    mov a,data_key
    anl a,#1110111b
    mov data_key,a
    Jb Row1,Column_3_Line_2
    Mov notujuan,#3
    ;Jnb Row1,$
    Acall LOnG_Delay
    Mov Data_Key,#0FFH
    Ret

```

```

Column_3_Line_2:
    mov a,data_key
    anl a,#1110111b
    mov data_key,a
    Jb Row2,Column_3_Line_3
    Mov notujuan,#6
    ;Jnb Row2,$
    Acall LOnG_Delay
    Mov Data_Key,#0FFH
    Ret

```

```

Column_3_Line_3:
    mov a,data_key
    anl a,#1110111b
    mov data_key,a
    Jb Row3,Column_3_Line_4

```

```

    Mov notujuan,#9
    ;Jnb Row3,$
    Acall L Ong_Delay
    Mov Data_Key,#0FFH
    Ret

Column_3_Line_4:
    mov a,data_key
    anl a,#11101111b
    mov data_key,a
    Jb Row4,Back
    mov notujuan,#11
    Mov A,#0Bh
    ;Jnb Row4,$
    Acall L Ong_Delay
    Mov Data_Key,#0FFH
    Ret

Long_Delay:
    Mov R7,#0FFh

Long_Delay_1:
    Acall Delay
    Djnz R7,Long_Delay_1
    Ret

Delay: Mov R6,#0FFh
    Djnz R6,$
    Ret

Sevseg:
    mov P1,notujuan
    mov P0,#0
    mov r0,#255
    mov r1,#25

ret

mulai: mov p0,#00000001b

```