

**RANCANG BANGUN SIMULASI SISTEM PENSINYALAN
LAYER TIGA PADA SISTEM KOMUNIKASI SELULER
BAGIAN *SUBSCRIBER* PENERIMA
MENGUNAKAN *MICROCONTROLLER* AT89S51**

SKRIPSI

OLEH

ULFA DWI UTAMI

04 04 03 082 2



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

**RANCANG BANGUN SIMULASI SISTEM PENSINYALAN
LAYER TIGA PADA SISTEM KOMUNIKASI SELULER
BAGIAN *SUBSCRIBER* PENERIMA
MENGUNAKAN *MICROCONTROLLER* AT89S51**

SKRIPSI

OLEH

ULFA DWI UTAMI

04 04 03 082 2



**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 SISTEM PENSINYALAN LAYER TIGA
PADA SISTEM KOMUNIKASI SELULER
BAGIAN *SUBSCRIBER* PENERIMA
MENGUNAKAN *MICROCONTROLLER AT89S51***

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, 15 Juli 2008

Ulfa Dwi Utami

NPM 04 04 03 082 2

PERSETUJUAN

Skripsi dengan judul :

**RANCANG BANGUN SIMULASI SISTEM PENSINYALAN LAYER TIGA
PADA SISTEM KOMUNIKASI SELULER
BAGIAN *SUBSCRIBER* PENERIMA
MENGUNAKAN *MICROCONTROLLER AT89S51***

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, 15 Juli 2008

Dosen Pembimbing

Dr. Abdul Muis, ST, M.Eng

NIP. 132 233 210

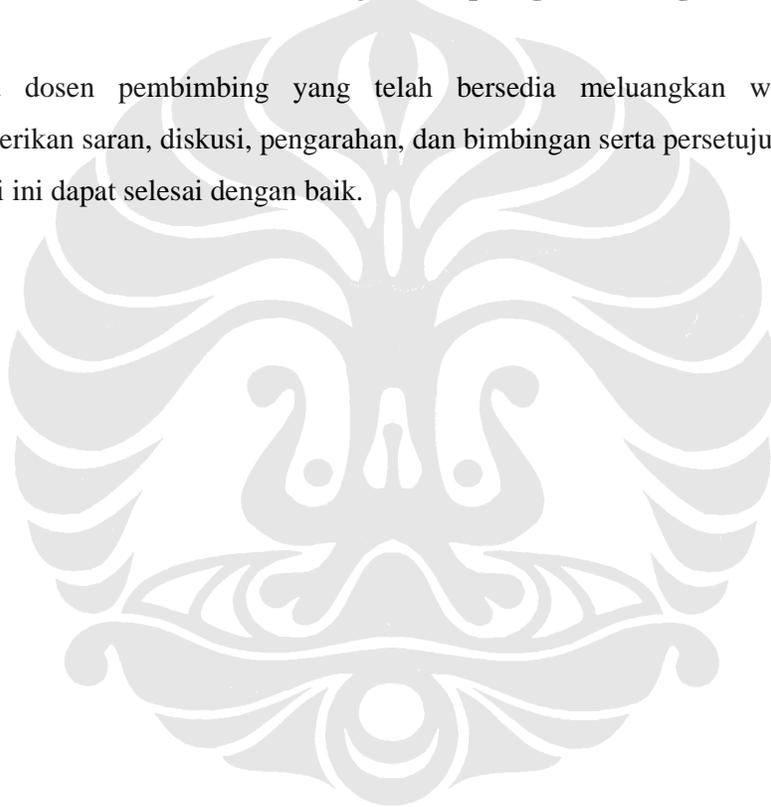
UCAPAN TERIMA KASIH

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

Dr. Abdul Muis, ST, M.Eng

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.



Ulfa Dwi Utami
NPM 04 04 03 082 2

Dosen Pembimbing
Dr. Abdul Muis, ST, M.Eng

Departemen Teknik Elektro

**RANCANG BANGUN SIMULASI SISTEM PENSINYALAN LAYER TIGA
PADA SISTEM KOMUNIKASI BERGERAK SELULER
BAGIAN *SUBSCRIBER* PENERIMA
MENGUNAKAN *MICROCONTROLLER AT89S51***

ABSTRAK

Pensinyalan merupakan bagian terpenting pada komunikasi seluler. Pensinyalan berperan sebagai pengontrol pada fungsi penyambungan saat pertukaran informasi. Bagian ini sangat diperlukan sebagai pengatur dari mulai pembuatan koneksi sampai dengan pengakhiran koneksi. Proses pengiriman pesan yang berupa aliran bit ini, merepresentasikan informasi pendukung yang mendukung pengiriman informasi intinya.

Pada skripsi ini dimodelkan sebuah rancangan sistem pensinyalan yang merepresentasikan sistem yang sebenarnya. Pensinyalan terjadi setiap kali MS melakukan koneksi dengan sentral. Pensinyalan yang dibahas pada MS penerima, meliputi kondisi *idle* dan menerima panggilan. Perancangan ini dimaksudkan untuk melihat bagaimana sistem pensinyalan bekerja serta mengetahui aliran bit dari dan menuju sentral.

Rancangan sistem tersebut mengacu pada sistem yang sudah ada, hanya saja terdapat beberapa penyesuaian. Penyesuaian tersebut dilakukan agar sistem dapat diaplikasikan menggunakan *microcontroller*. *Microcontroller* yang digunakan adalah tipe Atmel 89S51, yang memiliki 4 kbyte memori flash yang memungkinkan memori untuk diprogram ulang. Sebelum di aplikasikan ke dalam *microcontroller*, program tersebut diujikan pada perangkat lunak 8051 IDE.

Untuk pengujian sistem, simulator sistem pensinyalan tersebut, dihubungkan ke rangkaian *seven segment* dan rangkaian LED untuk menunjukkan bagaimana keluarannya menuju sentral. Pengujian dilakukan pada tiap rangkaian dan rangkaian sistem pensinyalan secara keseluruhan. Analisis dilakukan dengan melihat kinerja sistem. Aliran bit menuju sentral dan waktu yang digunakan untuk menjalankan sistem adalah parameter keberhasilan yang diamati. Kesimpulan yang dapat diambil adalah model sistem pensinyalan pada perancangan disini, sudah dapat merepresentasikan sistem yang ada. Namun tentu saja dengan beberapa penyesuaian agar dapat di aplikasikan menggunakan *microcontroller*.

Kata kunci : Komunikasi Seluler, Sistem Pensinyalan, *Microcontroller AT89S51*, Aliran Bit, 8051 IDE

Ulfa Dwi Utami
NPM 04 04 03 082 2

Counselor
Dr. Abdul Muis, ST, M.Eng

Electrical Engineering Departement

**SIMULATION SIGNALLING SYSTEM LAYER THREE PLANNING
ON MOBILE CELLULER COMMUNICATION ON RECEIVER
SUBSCRIBER DIVISION
WITH MICROCONTROLLER AT89S51**

ABSTRACT

Signaling is the most important part in cellular communication. On a switching function at information exchange, signaling is needed as a controller from the beginning until the end of connection. This process is a sending process of a message as flow of bits which represented information that support the main information.

In this paper, a system design is modeled to represent the real system. Signaling happens every time MS connecting with central. Signaling discussed are the signaling in MS receiver, which occur on idle and receiving calls condition. This design is meant to show how the signaling system works, also how are the flow of bit from and to central.

The system design referred to an existing system with a few adjustments. These adjustments are done so that the system can be applicable using a microcontroller. The microcontroller used is Atmel 89S51. It has 4 kbyte flash memory which enable reprogramming of the memory. Before applied in the microcontroller, this program is tested on a 8051 IDE software.

For the system testing, signaling system simulator is connected to a seven segment circuit and LED circuit to show how the output to central. The test is done on every circuit and the whole signaling system circuit. An analysis is taken from seeing the system performance. The flow of bit to central and total system time are the parameter to observe. The conclusion is this signaling model system design could represent the real system, but with a few adjustment for microcontroller application.

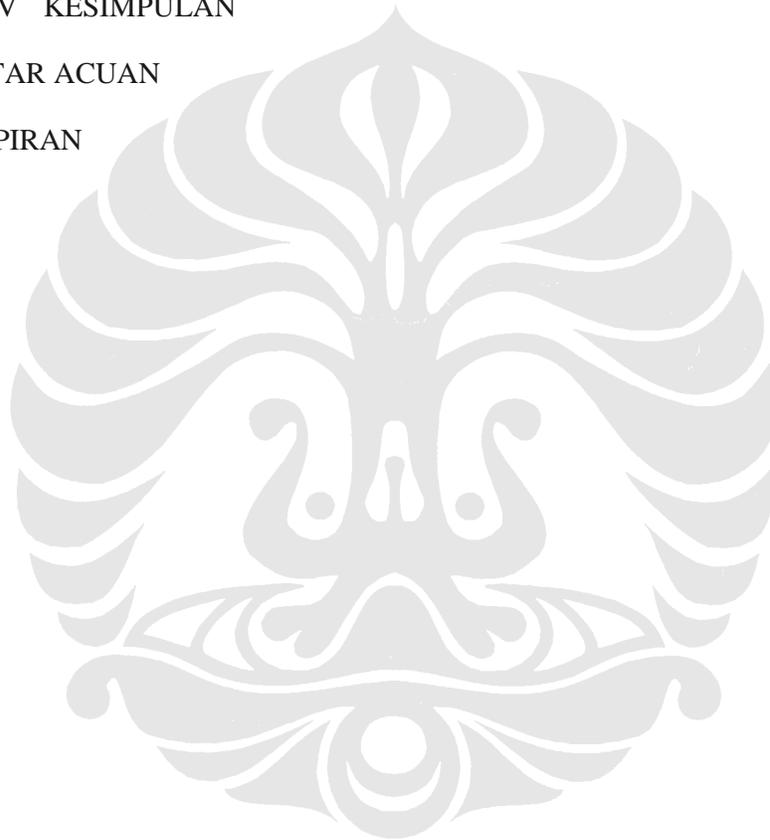
Keywords : Cellular Communication, Signalling System, *Microcontroller* AT89S51, Bit Flow, 8051 IDE

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	xi
DAFTAR LAMPIRAN	xii
DAFTAR SINGKATAN	xiii
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG	1
1.2. PERUMUSAN MASALAH	2
1.3. TUJUAN PENULISAN	2
1.4. BATASAN MASALAH	2
1.5. METODOLOGI PENULISAN	3
1.6. SISTEMATIKA PENULISAN	3
BAB II DASAR TEORI	4
2.1. PENSINYALAN PADA KOMUNIKASI SELULER	4
2.2. ELEMEN JARINGAN KOMUNIKASI SELULER	5
2.2.1. Interaksi Elemen Jaringan	6
2.2.1.1. <i>Kondisi Idle</i>	6
2.2.1.2. <i>Panggilan Masuk</i>	7
2.2.2. Penomoran Pada Jaringan Komunikasi Seluler [4]	9
2.2.2.1. <i>Location Area Identity (LAI)</i>	9
2.2.2.2. <i>International Mobile Equipment Identity (IMEI)</i>	10

2.2.2.3.	<i>International Mobile Subscriber Identity (IMSI)</i>	10
2.2.2.4.	<i>Temporary Mobile Subscriber Identity (TMSI)</i>	10
2.3.	PROTOKOL	11
2.3.1.	Arsitektur Protokol [10]	11
2.3.2.	Layer 3	11
2.3.2.1.	<i>CM</i>	11
2.3.2.2.	<i>MM</i>	12
2.3.2.3.	<i>RR</i>	12
2.4.	MICROCONTROLLER AT89S51	13
2.4.1.	Spesifikasi Sistem Minimum	13
2.4.2.	<i>Microcontroller AT89S51</i>	14
2.4.2.1.	<i>Blok Diagram</i>	15
2.4.2.2.	<i>Konfigurasi Pin Microcontroller [11]</i>	16
BAB III PERANCANGAN RANGKAIAN SIMULATOR SISTEM		
PENSINYALAN MENGGUNAKAN <i>MICROCONTROLLER AT89S51</i>		18
3.1	KONSEP PERANCANGAN RANGKAIAN SIMULATOR SISTEM PENSINYALAN	18
3.1.1.	Arsitektur Rangkaian Sistem PENSINYALAN	19
3.1.1.1.	<i>Rangkaian Port 0</i>	19
3.1.1.2.	<i>Rangkaian Port 2</i>	20
3.1.1.3.	<i>Rangkaian Port 3</i>	20
3.1.2.	Alur Permintaan dan Respon	21
3.1.2.1.	<i>Permintaan</i>	21
3.1.2.2.	<i>Respon Sentral</i>	21
3.2	FLOW CHART	22
3.2.1.	Kondisi <i>Idle</i>	22
3.2.1.1.	<i>IMSI attach</i>	23
3.2.1.2.	<i>Identifikasi ME</i>	24
3.2.1.3.	<i>Meng-update Lokasi</i>	24
3.2.2.	Panggilan Masuk	25
3.2.3.	Delay	28
3.2.4.	Sentral	29
BAB IV HASIL UJI COBA DAN ANALISIS PERANCANGAN SIMULASI		
SISTEM PENSINYALAN		30
4.1.	HASIL UJI COBA	30
4.1.1.	Software 8051 IDE	30
4.1.2.	Rangkaian Simulator Sistem PENSINYALAN	32
4.1.2.1.	<i>Pengujian Port 0</i>	34
4.1.2.2.	<i>Pengujian Port 1</i>	35

4.1.2.3. <i>Pengujian Port 3</i>	35
4.2. ANALISIS PERANCANGAN SISTEM PENSINYALAN	37
4.2.1. Permintaan	37
4.2.2. Respon Sentral	37
4.2.3. Penggunaan Tipe <i>Microcontroller</i>	38
4.2.4. Analisis Kejadian	38
4.2.4.1. <i>IMSI Attach</i>	39
4.2.4.2. <i>Identifikasi IMEI</i>	39
4.2.4.3. <i>Perbaruan Lokasi</i>	40
4.2.4.4. <i>Menerima Panggilan</i>	40
BAB V KESIMPULAN	42
DAFTAR ACUAN	43
LAMPIRAN	44

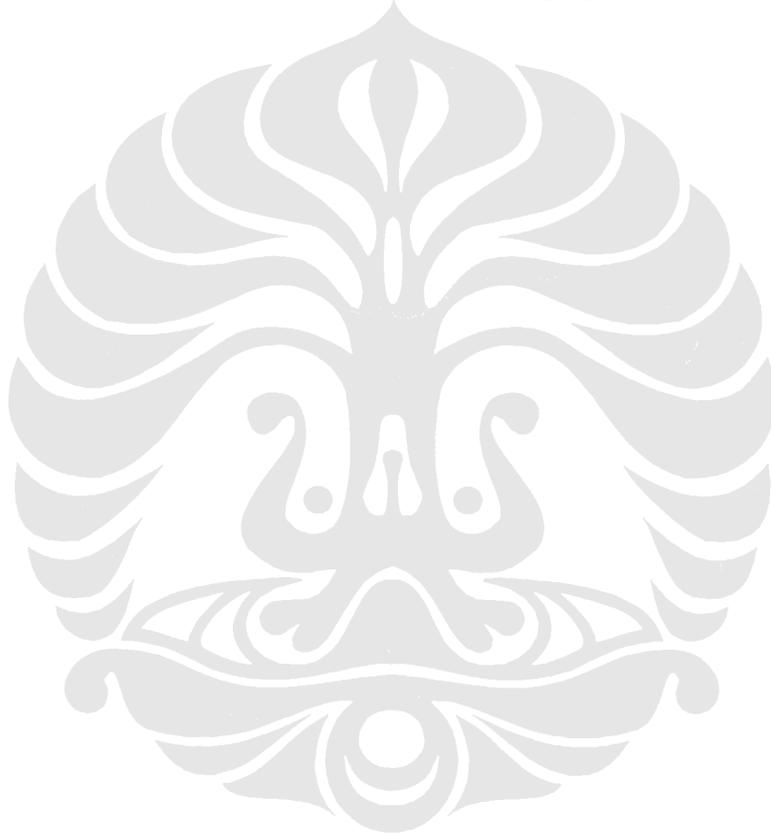


DAFTAR GAMBAR

	Halaman
Gambar 2.1. Arsitektur Jaringan (a) PSTN dan (b) PLMN [5]	4
Gambar 2.2. Elemen Jaringan GSM [6]	5
Gambar 2.3. Aliran Pesan saat melakukan Identifikasi ME [8]	6
Gambar 2.4. Aliran Pesan saat Perbaruan Lokasi [8]	7
Gambar 2.5. Aliran Pesan Panggilan Masuk [8]	9
Gambar 2.6. Arsitektur Protokol [10]	11
Gambar 2.7. Radio Interface Signalling	13
Gambar 2.8. Tata Letak Sistem Minimum LATIH 51P	14
Gambar 2.9. Blok Diagram <i>Microcontroller</i> AT89S51 [11]	15
Gambar 2.10. Konfigurasi Pin	16
Gambar 3.1. Rangkaian Simulator Sistem Pensinyalan	20
Gambar 3.2. <i>Flow Chart</i> Kondisi <i>Idle</i>	23
Gambar 3.3. Flow Chart Proses Panggilan Pada MS	26
Gambar 3.4. Penggambaran Saluran Menuju Sentral	29
Gambar 4.1. Hasil Pemeriksaan <i>Error</i>	30
Gambar 4.2. Tampilan Memori Internal pada 8051 IDE	31
Gambar 4.3. Tampilan Register pada 8051 IDE	32
Gambar 4.4. Tampilan Port pada 8051 IDE	32
Gambar 4.5. Tampilan Device Selection	33
Gambar 4.6. Tampilan saat Proses <i>Burning</i> Selesai	33
Gambar 4.7. Rangkaian Simulator Sistem Pensinyalan	34
Gambar 4.8. Rangkaian Port 0	34
Gambar 4.9. Rangkaian Port 1	35
Gambar 4.10. Rangkaian Port 3	35

DAFTAR TABEL

	Halaman
Tabel 3.1. Tabel Permintaan MS Penerima	21
Tabel 3.2. Tabel Respon Sentral	22
Tabel 4.1. Tabel Nilai Asumsi Sentral	31
Tabel 4.3. Tabel Perbedaan <i>Microcontroller</i> Atmel [13]	38



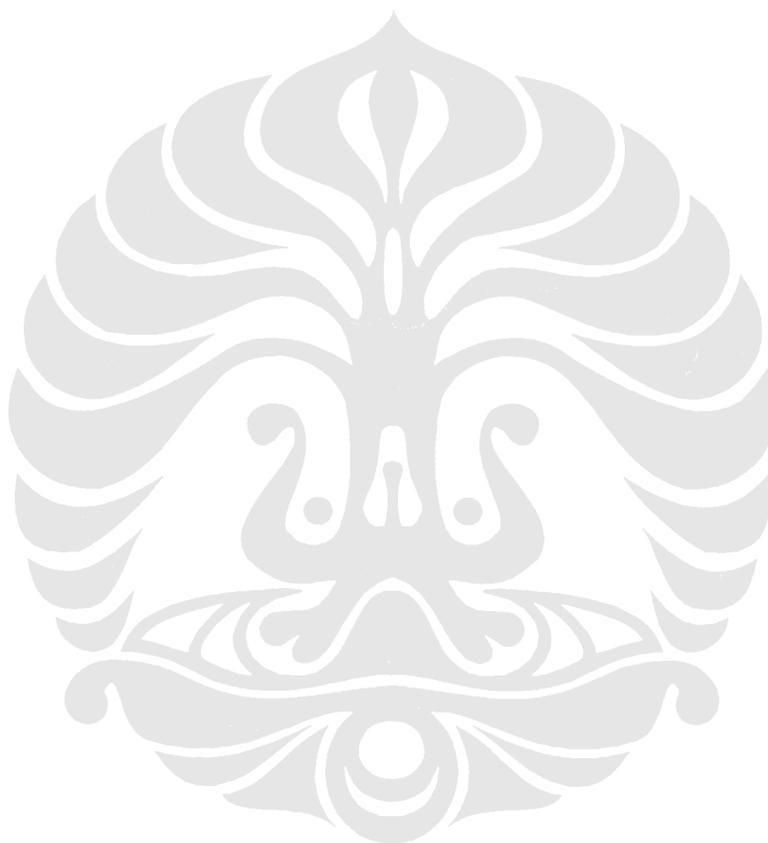
DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Listing Program Prosedur IMSI <i>Attach</i>	44
Lampiran 2 Listing Program Prosedur Identifikasi ME	45
Lampiran 3 Listing Program Prosedur Perbaruan Lokasi	45
Lampiran 4 Listing Program Prosedur Panggilan Masuk	46
Lampiran 5 Listing Program Prosedur <i>Delay</i>	47
Lampiran 6 Listing Program Sentral	48
Lampiran 7 Listing Program Deklarasi Variabel dan Deklarasi Alamat	50
Lampiran 8 Listing Program Nilai Pada Sentral	51
Lampiran 9 Listing Program Tes Port 0	52
Lampiran 10 Listing Program Tes Port 1	52
Lampiran 11 Listing Program Tes Port 3	52

DAFTAR SINGKATAN

BCCH	Broadcast Control Channel
BSS	Base Station Subsystem
BTS	Base Tranciever 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 Acces 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

Dalam beberapa dekade terakhir ini, penggunaan sistem telekomunikasi mengalami peningkatan yang sangat cepat. Sistem telekomunikasi yang paling diminati dan paling populer adalah sistem komunikasi seluler. [1] Sistem komunikasi ini memiliki keunggulan karena digunakan untuk memberikan layanan jasa telekomunikasi bagi pelanggan bergerak, sehingga dapat memenuhi kebutuhan komunikasi kapan saja, dimana saja, dan dengan siapa saja.

Di Indonesia sebagai negara berkembang, para pelaku dalam bidang seluler terus bersaing untuk mengembangkan teknologi baru guna menarik pelanggan sebanyak mungkin. Namun, yang sangat disayangkan, Indonesia sebagai salah satu negara yang konsumtif, belum mampu membuatnya sendiri. Negara kita hanya mampu membeli produk dari negara lain. Oleh karena itu, penulis mencoba membahas dan merancang bangun salah satu bagian terpenting dari komunikasi seluler, yaitu pensinyalan.

Konsep dasar pensinyalan pada komunikasi seluler yaitu sebagai fungsi pengontrolan pada fungsi penyambungan pertukaran informasi. Pensinyalan dikatakan sebagai pemberitahuan suatu *event* oleh *Mobile Station* (MS) ke sentral. Informasi pensinyalan dapat dibawa pada rute berbeda dari jalur yang membawa suara. [2] Pensinyalan diperlukan sebagai pengatur dari mulai pembuatan koneksi, pertukaran informasi sampai dengan pengakhiran koneksi.

Proses pensinyalan pada komunikasi seluler melibatkan banyak elemen jaringan, dari mulai MS sampai *Mobile Switching Center* (MSC). [3] Proses ini merupakan proses pengiriman pesan berupa aliran bit yang merepresentasikan informasi pendukung yang mendukung pengiriman informasi intinya. Sistem pensinyalan yang dibahas terbatas pada kondisi *idle* dan saat menerima panggilan.

Rancang bangun sistem pensinyalan ini menggunakan sistem minimum LATIH 51P versi 2.2 yang menggunakan satu *chip microcontroller* AT89S51

dengan 4Kbyte *memory*, memiliki jalur komunikasi serial UART RS-232, tersedia port *In-System Programming*. Sistem minimum tersebut diprogram sebagai MS penerima dan sebagai sentral untuk memberikan respon.

1.2. PERUMUSAN MASALAH

Masalah yang akan diteliti pada skripsi ini adalah rancang bangun simulasi sistem pensinyalan pada komunikasi seluler bagian *subscriber* penerima dengan menggunakan *microcontroller* AT89S51. Pemilihan *microcontroller* di sini dikarenakan aplikasi pada kenyataannya dulu menggunakan *microcontroller*, sehingga terlihat lebih mendekati keadaan yang sebenarnya. Selain itu aplikasi yang sederhana untuk memodelkan pensinyalan sebagai bagian terpenting dari pertukaran informasi untuk pertukaran informasi yang sebenarnya.

1.3. TUJUAN PENULISAN

Tujuan dari penulisan skripsi ini adalah sebagai berikut : (1) Penguasaan prinsip dasar perancangan sistem pensinyalan pada komunikasi seluler. (2) Mampu merancang bangun simulasi sistem pensinyalan pada komunikasi seluler menggunakan *microcontroller* AT89S51. (3) Menganalisis kinerja rangkaian simulasi sistem pensinyalan.

1.4. BATASAN MASALAH

Pembahasan dalam skripsi ini, dibatasi sebagai berikut : (1) Perancangan bangun sistem pensinyalan pada komunikasi seluler bagian *subscriber* penerima dengan menggunakan sistem minimum LATIH 51P versi 2.2 dengan sebuah *microcontroller* AT89S51 yang memiliki memori 4Kbyte. (2) Elemen jaringan yang terlibat hanya MS dan sentral (dalam hal ini MSC). (3) Aliran pesan pensinyalan terbatas pada aliran dari sentral ke MS saat kondisi *idle* dan saat menerima panggilan.

1.5. METODOLOGI PENULISAN

Penelitian dilakukan dengan membuat rancangan sistem pensinyalan menggunakan system minimum LATIH 51P versi 2.2 dengan *microcontroller* AT89S51 untuk diimplementasikan sebagai pensinyalan pada komunikasi seluler.

1.6. SISTEMATIKA PENULISAN

Skripsi ini terdiri atas 5 bab dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bagian ini membahas latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian dan sistematika penulisan.

BAB II DASAR TEORI

Menjelaskan dasar teori sistem pensinyalan pada komunikasi seluler, elemen-elemen jaringan yang terlibat di dalamnya, dan kejadian yang terjadi antara MS dengan sentral. Selain itu juga dijelaskan tentang spesifikasi *microcontroller* yang digunakan.

BAB III PERANCANGAN RANGKAIAN SIMULASI SISTEM PENSINYALAN MENGGUNAKAN MICROCONTROLLER AT89S51.

Menjelaskan arsitektur rangkaian simulasi sistem pensinyalan, diagram alir dan algoritma dasar perancangan rangkaian. Selain itu juga dijelaskan tentang representasi diagram alir dengan algoritma program.

BAB IV UJI COBA DAN ANALISIS RANGKAIAN SIMULASI SISTEM PENSINYALAN

Melakukan pengujian terhadap rangkaian yang dirancang, baik pengujian tiap bagian, maupun pengujian keseluruhan. Pengujian dilakukan menggunakan *software* maupun langsung terhadap rangkaian. Selain itu dianalisa perancangan sistem pensinyalan dari segi *hardware* maupun *software* serta diberikan solusi atas kendala yang dihadapi.

BAB V KESIMPULAN

Bagian ini berisi kesimpulan dari keseluruhan isi skripsi

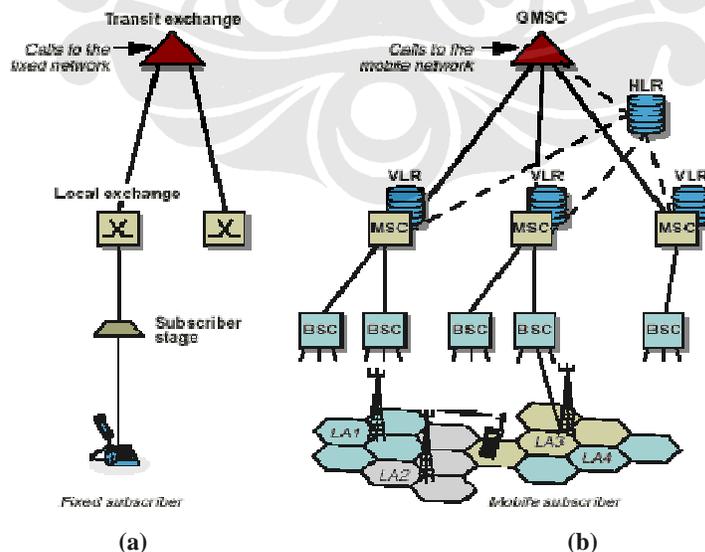
BAB II

DASAR TEORI

2.1. PENSINYALAN PADA KOMUNIKASI SELULER

Sistem komunikasi bergerak, atau yang biasa disebut dengan sistem komunikasi seluler digunakan untuk memberikan layanan jasa telekomunikasi bagi pelanggan bergerak. Keunggulan sistem ini adalah pelanggan mampu bergerak secara bebas di dalam area layanan sambil berkomunikasi tanpa terjadi pemutusan hubungan serta dapat dihubungi di mana saja pelanggan berada selama masih dalam daerah jangkauan jaringan yang digunakan. Pada sistem seluler, area layanan terbagi dalam beberapa daerah yang disebut sel. [4] Sel merupakan area jangkauan dari BTS (*Base Transceiver Station*), dimana setiap sel memiliki satu BTS yang dapat menjangkau seluruh area sel tersebut. [4]

Jaringan seluler atau PLMN (*Public Land Mobile Network*) terdiri dari sejumlah MS yang dihubungkan dengan jaringan radio ke infrastruktur perangkat *switching* yang terhubung dengan sistem lain seperti PSTN (*Public Switched Telephone Networks*). Gambar 2.1 memperlihatkan perbandingan jaringan PLMN dengan jaringan PSTN.

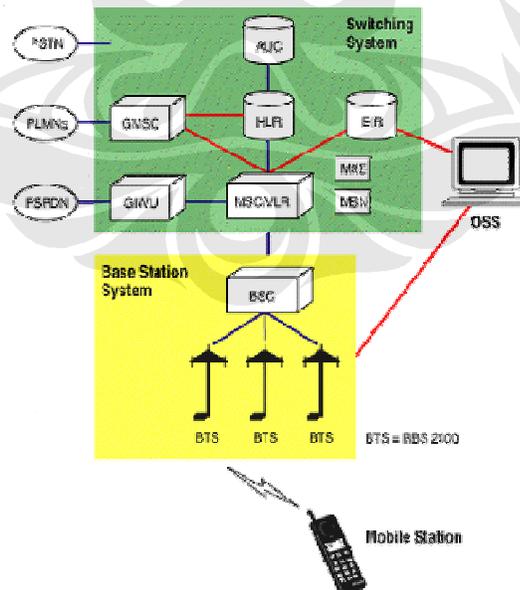


Gambar 2.1. Arsitektur Jaringan (a) PSTN dan (b) PLMN [5]

Pensinyalan pada sistem komunikasi seluler terjadi setiap kali MS melakukan koneksi dengan sentral. Pensinyalan dilakukan oleh MS berdasarkan perintah dari pengguna. Saat MS pertama dinyalakan, terjadi tiga pensinyalan penting pada kondisi *idle*, yaitu identifikasi ME (*Mobile Equipment*), *cell reselection*, dan *location area update*. Selain itu pensinyalan juga dilakukan saat MS melakukan atau menerima panggilan. Jadi, pensinyalan merupakan ‘pembuka jalan’ MS melakukan koneksi dengan sentral.

2.2. ELEMEN JARINGAN KOMUNIKASI SELULER

Secara umum sistem jaringan komunikasi seluler meliputi tiga subsistem utama yang saling terhubung dan saling berinteraksi. Yang termasuk ke dalam subsistem adalah *Base Station Subsystem* (BSS), *Network and Switching Subsystem* (NSS), dan *Operation Support Subsystem* (OSS). MS juga termasuk sebagai sebuah subsistem, tapi umumnya digolongkan ke bagian dari BSS untuk alasan arsitektur. [6] Gambar 2.2 menampilkan struktur jaringan GSM (*Global System for Mobile Communication*), yang terdiri dari elemen-elemen pendukung setiap subsistem.



Gambar 2.2. Elemen Jaringan GSM [6]

2.2.1. Interaksi Elemen Jaringan

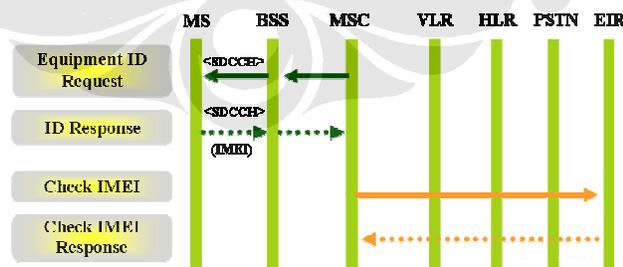
Dalam menyediakan sebuah layanan diperlukan kerja sama yang sinergis antara elemen jaringan tersebut, karena setiap elemen mempunyai fungsi yang spesifik. Elemen-elemen tersebut akan bekerja sama selama penyediaan, pengaturan maupun pelepasan panggilan.

Proses dasar yang dilakukan MS adalah pendaftaran lokasi dan *me-routing* panggilan. Pendaftaran lokasi menjadi persyaratan dasar agar setiap panggilan dapat mencapai pelanggan yang *roaming*. Untuk kepentingan kontrol, wilayah cakupan geografis jaringan nasional dibagi ke dalam sejumlah LA (*Location Area*). Informasi dan kontrol dari setiap LA diberikan oleh BCCH (*Broadcast Control Channel*).

2.2.1.1. Kondisi Idle

Kondisi *idle* adalah keadaan dimana MS dalam keadaan *standby* dan tidak sedang melakukan panggilan. Pada mode ini, penting untuk MS untuk dapat diakses dan dijangkau oleh sistem. Perilaku mode *idle* diatur oleh MS. Saat MS dalam keadaan *idle* akan terdapat beberapa kejadian umum seperti yang akan diuraikan di bawah ini. [7]

Identifikasi ME terjadi saat pertama kali MS dinyalakan. Proses ini untuk melihat apakah ME yang digunakan berada pada kategori ME yang bisa dilayani atau tidak. Tahap-tahap kejadian saat identifikasi ME ditunjukkan pada Gambar 2.3.

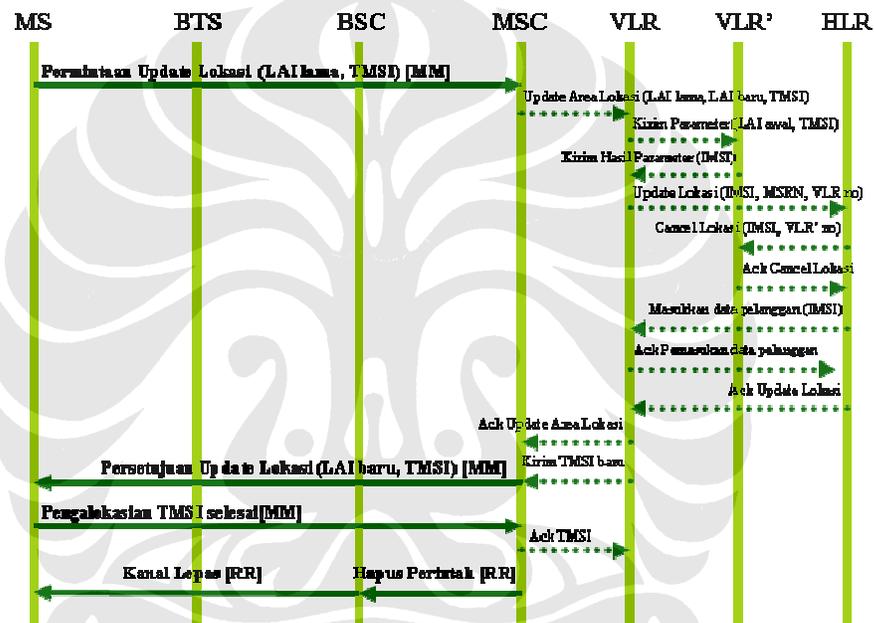


Gambar 2.3. Aliran Pesan saat melakukan Identifikasi ME [8]

Cell reselection, atau biasa disebut dengan IMSI (*International Mobile Subscriber Identity*) *attach* dilakukan untuk menentukan sel yang *servicing* ke 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 kembali sel, atau biasa disebut *cell reselction*. [7]

Saat melakukan pensinyalan dengan sentral, MS tidak selalu berada pada satu area jangkauan suatu BTS, oleh karena itu ada prosedur yang dinamakan *location area update*. Perbaruan lokasi ini dengan melihat LAI pada MS, jika sudah tidak sama, maka MS akan meng-*update* lokasi. Dengan prosedur ini, informasi lokasi MS pada jaringan akan selalu diperbarui. Pada Gambar 2.4. ditunjukkan tahap-tahap ketika melakukan *location area update*.



Gambar 2.4. Aliran Pesan saat Perbaruan Lokasi [8]

2.2.1.2. Panggilan Masuk

Proses dasar yang kedua adalah *routing* panggilan. Panggilan adalah kondisi dimana MS akan melakukan pertukaran informasi, baik dengan MS lain maupun dengan PSTN. Dalam skripsi ini, penulis hanya membahas tentang proses panggilan masuk. Terdapat tiga tahap utama saat menerima suatu panggilan.

Tahap pertama, menghubungi GMSC (*Gateway MSC*). Ketika seseorang dari PSTN melakukan panggilan ke MS, mereka menekan nomor telepon (MSISDN) dan panggilan akan di-*routing* ke operator

GMSC. [9] GMSC berfungsi sebagai jalan masuk dari bagian luar PLMN (PSTN atau PLMN lain) ke jaringan PLMN tersebut. GMSC digunakan untuk menentukan lokasi saat ini dari MS untuk menyambungkan panggilan. GMSC melakukan ini dengan berkonsultasi dengan HLR (*Home Location Register*).

Tahap kedua, *routing* panggilan. Ketika HLR menerima pesan meragukan, HLR akan menentukan apakah panggilan harus di-*routing* ke nomor lain (dialihkan), atau 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 me-*routing* ke nomor CFU. Jika MS tidak secara langsung terhubung dengan VLR (*Visitor Location Register*) 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. [9]

Akhirnya, jika HLR mengetahui bahwa MS tersebut berada dalam tanggung jawab terutama dari VLR, maka MS akan meminta nomor sementara (MSRN) dari VLR tersebut. Nomor ini disampaikan ke GMSC, yang digunakan untuk me-*routing* panggilan ke MSC yang dikunjungi. [9]

Tahap ketiga, membunyikan telepon. Saat panggilan diterima oleh MSC yang dikunjungi, MSRN (*Mobile Station Roaming Number*) digunakan untuk menemukan lokasi MS pada VLR. [9] *Paging* terjadi pada semua BTS di area tersebut. Ketika MS pelanggan merespon, lokasi dari MS tujuan dikembalikan ke MSC yang dikunjungi. VMSC (*Visit MSC*) lalu meneruskan panggilan ke BTS yang semestinya, dan MS tujuan akan berbunyi. Jika pelanggan menjawab, kanal suara akan terbentuk melewati VMSC selanjutnya GMSC kembali ke jaringan dimana pelanggan yang membuat panggilan. [9]

Jika MS tujuan tidak menjawab, karena pelanggan sedang sibuk dengan nomor lain (*call waiting* tidak digunakan), VMSC me-*routing* panggilan ke nomor CFB (*Call Forward Busy*). [9] Demikian pula, jika pelanggan tidak menjawab panggilan setelah beberapa periode waktu

kode pembeda setiap sistem jaringan dalam satu negara. Sedangkan LAC (*Location Area Code*) merupakan kode dari suatu sel dimana MS tersebut berada.

2.2.2.2. *International Mobile Equipment Identity (IMEI)*

IMEI menyatakan identitas yang jelas dari perangkat yang digunakan sebagai administrasi perangkat yang tersimpan di EIR (*Equipment Identity Register*). Format penomorannya adalah sebagai berikut.

TAC + FAC + SNR + SP

Nomor IMEI terdiri dari 15 digit, 6 digit pertama merupakan TAC (*Type Approval Code*), 2 digit selanjutnya FAC (*Final Assembly Code*), lalu SNR (*Serial Number*) pada 6 digit kemudian, dan 1 digit terakhir merupakan SP (*Substitute Position*).

2.2.2.3. *International Mobile Subscriber Identity (IMSI)*

IMSI menyatakan kode identitas yang unik dari pelanggan suatu jaringan. IMSI disimpan pada SIM dan HLR. Formatnya memiliki panjang maksimal 15 angka. Berikut format penomorannya.

MCC + MNC + HLR ID + Subscriber Number

$n_1n_2n_3 \quad n_4n_5 \quad n_6n_7 \quad n_8n_9n_{10}n_{11}n_{12}n_{13}n_{14}n_{15}$

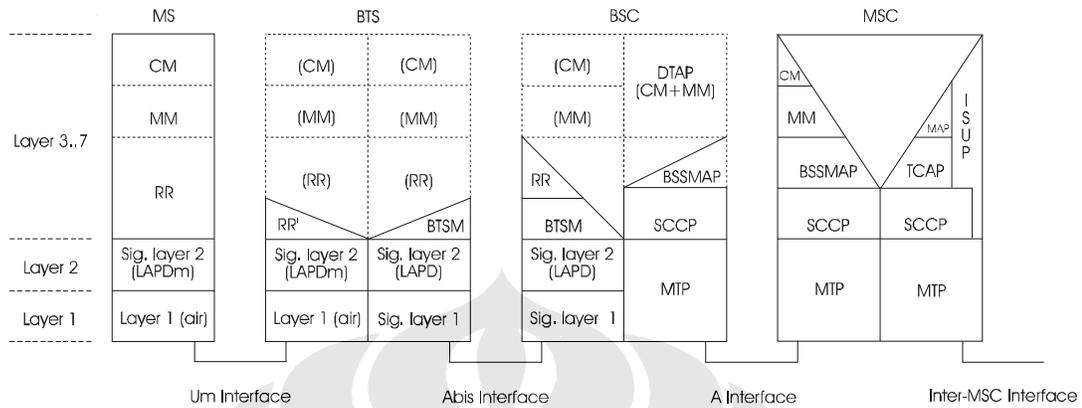
MCC untuk Indonesia kodenya adalah 510. Kode MNC merupakan kode operator dari setiap jaringan, kode 01 dan 21 untuk Indosat, 10 untuk Telkomsel, dan kode 11 untuk XL.

2.2.2.4. *Temporary Mobile Subscriber Identity (TMSI)*

TMSI menyatakan nomor identitas temporer yang unik dari *subscriber* sebagai pengganti IMSI yang hanya digunakan di 1 MSC. IMSI bernilai random dengan panjang 32 bit.

2.3. PROTOKOL

2.3.1. Arsitektur Protokol [10]



Gambar 2.6. Arsitektur Protokol [10]

Arsitektur protokol jaringan komunikasi seluler diperlihatkan pada Gambar 2.6. Setiap elemen jaringan memiliki 7 layer, dimana masing-masing memiliki protokol dan tugasnya tersendiri.

Layer 1 (*Physical Layer*) berupa hubungan fisik, elektrik, dan karakteristik fungsional dari pensinyalan layer 2, atau dengan kata lain layer ini sebagai alat untuk mengaksesnya.

Layer 2 (*Data Link Layer*) berisi fungsi dan pelaksanaan prosedur untuk mengontrol pengiriman informasi pensinyalan melalui sebuah *data link*, seperti pembatasan frame, pendeteksi kesalahan, pengkoreksi kesalahan dengan mengirim ulang, pengontrol aliran link, dll.

Layer 3 (*Network Layer*) meliputi semua prosedur pensinyalan yang dapat dibagi sebagai berikut, yaitu *Call Management*, *Mobility Management*, dan *Radio Resource Management*.

2.3.2. Layer 3

2.3.2.1. CM

CM dapat digunakan untuk kontrol panggilan (*Call Control*), pengelolaan layanan tambahan (*Supplementary Services*), dan pengelolaan layanan pesan singkat (*Short Message Service*). CC bertanggung jawab pada proses membuat, menjaga, dan mengakhiri panggilan. Salah satu

fungsi CM yang paling penting adalah *call routing* dalam rangka untuk mencari MS pelanggan.[10]

Pesan-pesan yang ditangani oleh CC adalah *alerting*, proses panggilan seperti *set up* panggilan, koneksi, konfirmasi panggilan, persetujuan koneksi, sampai pengakhiran panggilan. Sedangkan SS menangani *register SS*, penghapusan SS, pengaktifan serta penonaktifan SS, *register password*, dan sebagainya.

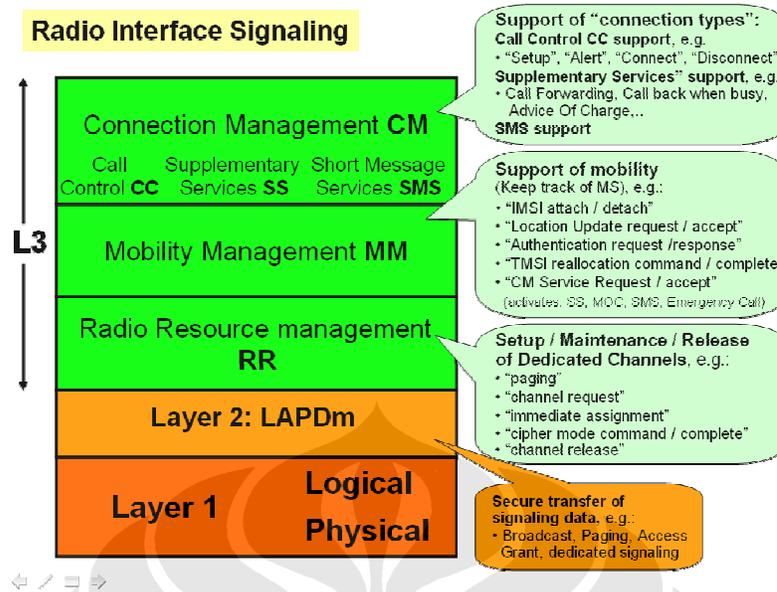
2.3.2.2. MM

Fungsi MM berkaitan dengan semua aspek yang terkait dengan mobilitas pengguna, terutama pengelolaan lokasi serta autentikasi dan keamanannya. [10] Prosedur autentikasi melibatkan kartu SIM dan pusat autentikasi (*Authentication Center*). Pada MS dan jaringannya sama-sama menyimpan kode rahasia (Ki). Dengan menggunakan A3, dari Ki dan RAND akan menghasilkan SRES (*Signed Response*). Perhitungan SRES di SIM dan AuC ini yang akan dibandingkan.

Pesan lainnya yang ditangani oleh MM adalah indikasi IMSI detach, permintaan dan respon identitas, persetujuan dan penolakan location update, permintaan location update, perintah relokasi TMSI, permintaan, penolakan dan respon autentikasi dan lain-lain.

2.3.2.3. RR

Fungsi RR berkaitan dengan membangun, menjaga, dan mengakhiri komunikasi antara MS dengan MSC. [10] 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*.



Gambar 2.7. Radio Interface Signalling

Gambar 2.7. di atas menampilkan pensinyalan *radio interface*, layer 1 sampai layer 3, serta fungsi-fungsinya pada sistem pensinyalan komunikasi seluler.

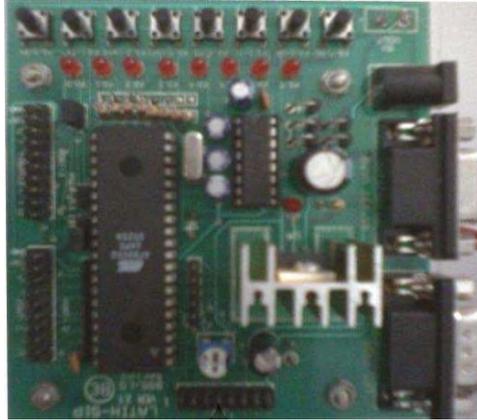
2.4. MICROCONTROLLER AT89S51

2.4.1. Spesifikasi Sistem Minimum

Sistem minimum (sismin) *microcontroller* adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC *microcontroller*. Sismin ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu. Board sistem minimum yang digunakan adalah LATIH 51P ver. 2.2. Sismin ini cocok untuk aplikasi sederhana. Beberapa fitur sismin ini yaitu :

- ◆ Mendukung *microcontroller* AT89S51, AT89S52, AT89S53, AT89S8252,
- ◆ Terdapat 8 LED *onboard*,
- ◆ Terdapat 8 *push button onboard*,
- ◆ Dilengkapi post ISP, port LCD, dan Port Serial RS232,

Gambar 2.8 berikut menunjukkan tata letak sistem minimum LATIH 51P.



Gambar 2.8. Tata Letak Sistem Minimum LATIH 51P

2.4.2. *Microcontroller AT89S51*

AT89S51 mempunyai konsumsi daya rendah, *microcontroller* 8-bit CMOS dengan 4K byte memori *Flash ISP (in-system programmable)* yang artinya dapat diprogram didalam sistem. [11] Divais ini dibuat dengan teknologi memori *non-volatile* kerapatan tinggi dan cocok dengan semua instruksi dan pin keluaran 8051. Dengan sifat *Non-volatile ini*, memori yang akan kehilangan informasinya ketika tidak mendapatkan daya. [12]

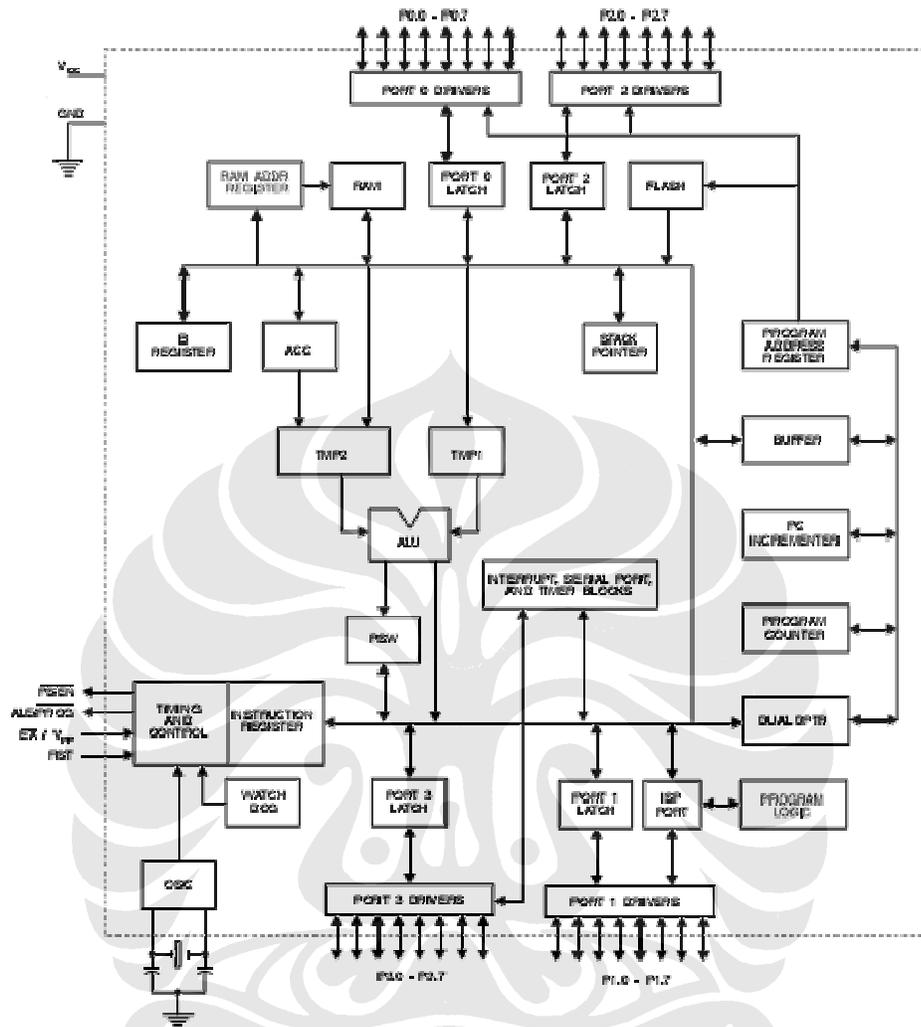
Flash yang berada didalam *chip* memungkinkan memori untuk diprogram ulang pada saat *chip* didalam sistem. Dengan mengkombinasikan CPU 8 bit yang serbaguna dengan *flash ISP* pada *chip*, AT89S51 merupakan *microcontroller* yang memberikan fleksibilitas yang tinggi dan biaya yang murah. [11]

AT89S51 memberikan beberapa fitur standar, yaitu :

- ◆ 128 byte RAM,
- ◆ 32 saluran I/O,
- ◆ Dua data pointer,
- ◆ Dua 16-bit pewaktuan,
- ◆ Port serial yang *full duplex*,
- ◆ Pembangkit gelombang (*oscillator*) yang *on-chip*.

Sebagai tambahan, *microcontroller* ini memiliki ketahanan untuk diprogram ulang hingga 1000 kali.

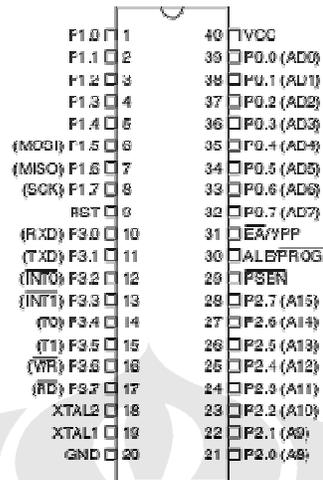
2.4.2.1. Blok Diagram



Gambar 2.9. Blok Diagram *Microcontroller* AT89S51 [11]

Gambar 2.9. diatas menampilkan blok diagram dari *microcontroller* AT89S51. *Microcontroller* ini memiliki 4 port untuk input keluaran data, akumulator, *register*, *RAM*, *stack pointer*, *Arithmetic Logic Unit* (ALU), pengunci (*latch*) dan rangkaian osilasi yang membuat 89C51 dapat beroperasi hanya dengan 1 keping IC.

2.4.2.2. Konfigurasi Pin Microcontroller [11]



Gambar 2.10. Konfigurasi Pin

AT89S51 memiliki 40 pin dengan konfigurasi seperti terlihat pada Gambar 2.10. di atas. Berikut penjelasan beberapa pin pada AT89S51.

Port 0 Port 0 adalah 8 bit *open drain bi-directional* port I/O. Pada saat sebagai port keluaran, tiap pin dapat dilewatkan ke-8 input TTL. Ketika logika satu dituliskan pada port 0, maka pin-pin ini dapat digunakan sebagai input berimpedansi tinggi. Port 0 dapat dikonfirmasi untuk *demultiplex* sebagai jalur data atau alamat bus selama membaca ke program eksternal dan memori data. Pada mode ini P0 mempunyai internal *Pullup*. Port 0 juga menerima kode byte selama pemrograman *Flash* dan mengeluarkan kode byte selama verifikasi program.

Port 1 Port 1 adalah 8 bit *bi-directional* port I/O dengan internal *Pullup*. Port 1 mempunyai keluaran yang dapat dihubungkan dengan 4 TTL input. Ketika logika '1' dituliskan ke port 1, pin ini di *pull high* dengan menggunakan internal *pullup* dan dapat digunakan sebagai input.

Port 2 Port 2 adalah 8 bit *bi-directional* port I/O dengan *Pullup*. Port 2 keluaran *buffer* dapat melewatkan empat TTL input. Ketika logika satu dituliskan ke port 2, maka mereka di-*pull*

high dengan internal *Pullup* dan dapat digunakan sebagai input.

Port 3 Port ini juga 8 bit *bi-directional* port I/O dengan *Pullup*. Keluaran buffer dari Port 3 dapat dilewati empat input TTL. Ketika logika satu dituliskan ke port 3, maka mereka akan di-*pull high* dengan internal *pullup* dan dapat digunakan sebagai input. Port 3 juga menerima beberapa sinyal kontrol untuk pemrograman *Flash* dan verifikasi.

RST Input *reset*. Logika *high* pada pin ini akan *reset* siklus mesin.

ALE/PROG Pulsa keluaran ALE (*Address Latch Enable*) digunakan untuk mengunci byte bawah dari alamat selama mengakses ke memori eksternal. Jika dikehendaki, operasi ALE dapat di-*disable* dengan memberikan *setting* bit 0 dari SFR pada lokasi 8EH.

PSEN *Program Store Enable* merupakan sinyal yang digunakan untuk membaca program memori eksternal. Ketika 8951 mengeksekusi kode dari program memori eksternal, PSEN diaktifkan dua kali setiap siklus mesin.

EA/VPP *Eksternal Acces Enable*, EZ harus diposisikan ke GND untuk mengaktifkan divais untuk mengumpankan kode dari program memory yang dimulai pada lokasi 0000h sampai FFFFh. EA harus diposisikan ke VCC untuk eksekusi program internal.

XTAL1 Input untuk *oscillator inverting amplifier* dan input untuk internal *clock* untuk pengoperasian rangkaian.

XTAL2 Keluaran dari *inverting oscillator amplifier*.

BAB III

PERANCANGAN RANGKAIAN SIMULATOR SISTEM PENSINYALAN MENGGUNAKAN *MICROCONTROLLER AT89S51*

3.1 KONSEP PERANCANGAN RANGKAIAN SIMULATOR SISTEM PENSINYALAN

Pada perancangan rangkaian simulator sistem pensinyalan ini, terdapat beberapa tahap yang harus dilakukan. Beberapa tahap sudah dilakukan pada studi sebelumnya. Tahap yang sudah dilakukan yaitu membuat *flow chart* berdasarkan *message flow* hasil studi pustaka dari sebuah perusahaan vendor di Indonesia. Studi pustaka tersebut dilakukan agar simulator yang dirancang mendekati sistem pensinyalan yang sebenarnya. Selain juga dibuat algoritma yang merepresentasikan setiap blok pada *flow chart* yang sudah ada.

Tahap selanjutnya yang dilakukan pada skripsi ini diantaranya membuat program dari algoritma yang sudah ada menggunakan bahasa *assembly*. Untuk memeriksa kebenaran program tersebut, maka program disimulasikan menggunakan *software* simulator 8051 IDE sampai tidak ada *error* yang terjadi.

Untuk *hardware*, simulator sistem pensinyalan ini, dibuat dengan menggabungkan beberapa rangkaian terpisah, yaitu sistem minimum LATIH 51P versi 2.2, rangkaian *seven segment* dan 8 resistor $1k\Omega$ yang terhubung dengan port 0, rangkaian 8 LED yang terhubung dengan port 2, serta rangkaian *push button* yang terhubung dengan port 3.

Sistem minimum LATIH 51P versi 2.2 merupakan satu modul *In-System Programming* dengan sebuah *chip microcontroller AT89S51*. Namun, untuk memprogram *chip* ini, penulis melakukan pemrograman melalui kabel RS232. Sistem minimum ini memiliki keuntungan dapat bekerja sendiri (*stand alone*), artinya setelah program di-*download* ke *board* LATIH 51P, maka dapat langsung dijalankan tanpa perlu menambahkan atau mengganti komponen. Selain itu sistem

minimum ini sudah dilengkapi 8 LED *onboard* pada port 2 dan 8 *push button onboard* pada port 3.

Seperti sudah dijelaskan pada sub sub bab 2.4.2, *microcontroller* yang digunakan memiliki beberapa keuntungan dibandingkan dengan *microcontroller* versi sebelumnya, yaitu konsumsi daya yang rendah, memiliki memori *Flash ISP* sehingga program dapat diprogram didalam sistem, memiliki sifat *non-volatile*, serta fleksibilitas yang tinggi dengan biaya yang murah. Selain itu, *microcontroller* ini memiliki ketahanan diprogram ulang hingga 1000 kali. [11]

Dalam merancang rangkaian sistem pensinyalan, penulis mengacu pada sistem yang sudah ada, hanya saja terdapat beberapa penyesuaian sehingga sistem ini dapat diaplikasikan menggunakan *microcontroller*. Penyesuaian ini diperlukan karena terdapat beberapa keterbatasan pada jenis *microcontroller* yang digunakan, diantaranya adalah memori program yang terbatas, sehingga penulis hanya memasukkan kejadian-kejadian yang umum terjadi saja. Selain ini dikarenakan *microcontroller* yang digunakan hanya satu, maka pada memori program *microcontroller* tersebut akan terdapat program pada sentral yang berfungsi untuk memberikan respon pada *microcontroller* tersebut yang berfungsi sebagai *subscriber* penerima. Penyesuaian yang dilakukan tetap tidak menghilangkan prinsip dasar dari pensinyalan pada komunikasi seluler. Penulis tetap menggunakan data yang didapat dari hasil *tracing* salah satu vendor di Indonesia yang menunjukkan nilai bit yang digunakan untuk menjalankan suatu perintah.

3.1.1. Arsitektur Rangkaian Sistem Pensinyalan

Seperti sudah dijelaskan pada sub bab 3.1, rangkaian pensinyalan ini memiliki tiga rangkaian terpisah yang terhubung pada tiga port *microcontroller*.

3.1.1.1. Rangkaian Port 0

Rangkaian terdiri dari satu *seven segment*, dan 8 resistor 1 k Ω . Rangkaian ini berfungsi untuk melihat bit keluaran perintah menuju sentral, baik permintaan dari MS maupun nilai yang di kirim ke sentral, seperti nilai IMSI, IMEI, dan nilai LAI.

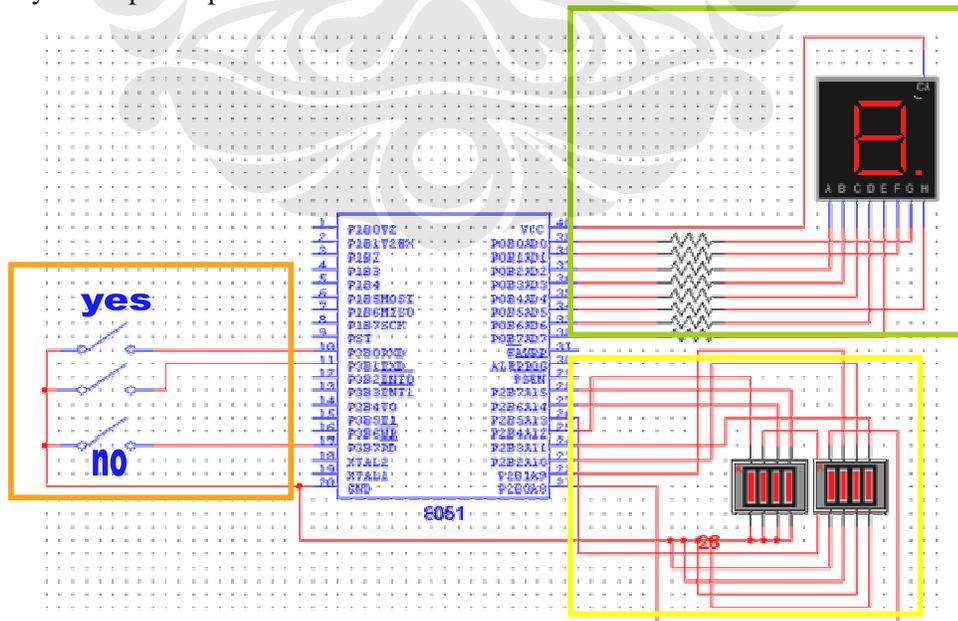
3.1.1.2. Rangkaian Port 2

Port 1 memperlihatkan rangkaian dengan delapan LED. Rangkaian ini berfungsi sebagai indikator adanya panggilan masuk dari sentral. Ketika sentral mengirimkan permintaan *alerting*, maka LED akan terus menyala sampai ada perintah untuk mematikan dari port 3.

3.1.1.3. Rangkaian Port 3

Rangkain port 3 berada pada sistem minimum yang digunakan. Pada rangkaian ini hanya mengaktifkan dua *push button* yang ada yang pada aplikasinya berfungsi sebagai *keypad* 'yes' dan 'no' pada ME. Ketika ada indikasi panggilan, dengan menekan tombol 'yes' maka panggilan akan terhubung, sedangkan jika menekan tombol 'no', maka panggilan akan berakhir. *Push button* 1 yang berfungsi sebagai tombol 'yes' dihubungkan dengan pin 3.0 sedangkan *push button* 2 yang berfungsi sebagai tombol 'no' dihubungkan dengan pin 3.7.

Gambar 3.1 dibawah ini, menunjukkan gambar rangkaian secara keseluruhan yang terhubung dengan microcontroller AT89S51. Pada kotak warna hijau merupakan port 0, kotak warna kuning merupakan port 2, dan kotak warna oranye merupakan port 3.



Gambar 3.1. Rangkaian Simulator Sistem Pensinyalan

3.1.2. Alur Permintaan dan Respon

Rangkaian sistem pensinyalan ini selain berfungsi sebagai MS penerima, juga berfungsi sebagai sentral. Di dalamnya seperti terdapat saluran sentral yang di buat pada memori internal 60h-69h. Saluran sentral di sini dibuat untuk memberikan respon terhadap permintaan di port 1.

3.1.2.1. Permintaan

Alur permintaan pada MS penerima di sesuaikan dengan data yang penulis dapatkan. Saat kondisi *idle* terdapat beberapa permintaan diantaranya permintaan untuk mengalokasikan imsi, permintaan nilai imei, serta permintaan pembaruan lokasi. Sedangkan pada saat MS menerima panggilan, terdapat beberapa permintaan diantaranya permintaan pemakaian kanal, permintaan layanan, konfirmasi panggilan, serta pemberitahuan pengakhiran panggilan dapat dilakukan. Pada tabel 3.1. ditunjukkan permintaan yang dilakukan oleh MS penerima.

Tabel 3.1. Tabel Permintaan MS Penerima

Permintaan		Besar Bit Biner	Tampilan
Lambang	Penjelasan		
imsi_aloc	Alokasi imsi	#00000010b	2
akhir_tmsi	Selesai alokasi tmsi	#00110011b	-
imei	Permintaan nilai imei	#00000101b	-
update_lok_req	Pengecekan nomor LAI	#00000111b	7
up_lok_req	Permintaan pembaruan lokasi	#00001000b	8
kanal_req	Permintaan kanal	#00000001b	1
service_req	Permintaan layanan	#00000101b	5
konf_call	Konfirmasi panggilan	#00001001b	9
busy	Pemberitahuan sibuk	#10101010b	3
call_rls_ok	Pengakhiran panggilan dapat dilakukan	#00000110b	6

3.1.2.2. Respon Sentral

Alur respon sentral di sesuaikan permintaan yang MS lakukan. Setiap permintaan memiliki responnya masing-masing. Respon yang diberikan seharusnya diberikan dari sentral, dalam hal ini MSC. Pada tabel 3.2. ditunjukkan respon sentral beserta besar bit binernya.

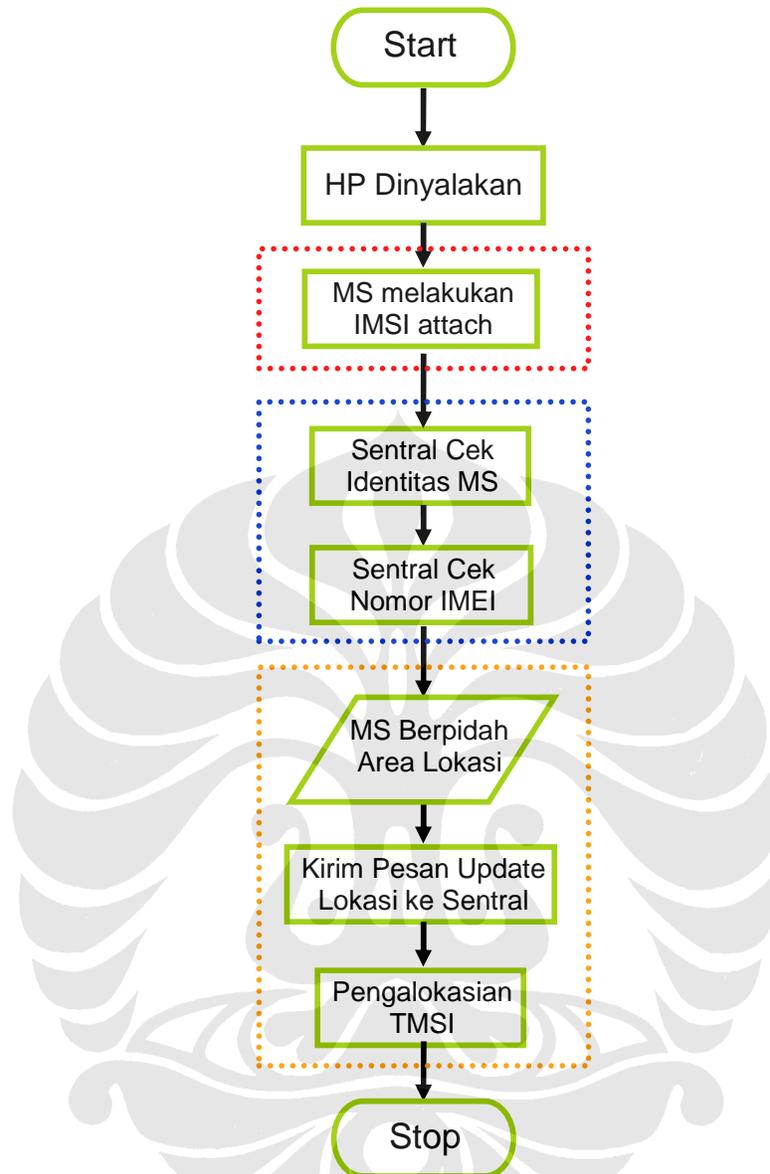
Tabel 3.2. Tabel Respon Sentral

Lambang Respon Sentral	Respon Permintaan	Besar Bit Biner
imsi_aloc_app	imsi_aloc	#11110000b
aloc_tmsi_fnsh	akhir_tmsi	#01001011b
iden_req	imei	#00011000b
LAI_req	update_lok_req	#00011001b
LAI_aloc	up_lok_req	#00000100b
pag_req	standby	#10000001b
kanal_req_app	kanal_req	#00000011b
service_req_app	service_req	#00100100b
alert_req	konf_call	#11111111b

3.2 FLOW CHART

3.2.1. Kondisi *Idle*

Gambar *flow chart* pada kondisi *idle* ditunjukkan pada Gambar 3.2 dibawah ini. *Flow chart* yang didalam kotak dengan garis putus-putus berwarna merah merupakan prosedur penentuan BTS yang *servicing* ke MS tersebut, atau disebut *IMSI attach*. Kemudian *flow chart* yang didalam kotak dengan garis putus-putus berwarna biru adalah prosedur identifikasi ME yang digunakan. Sedangkan *flow chart* yang berada dalam kotak dengan garis putus-putus berwarna kuning adalah prosedur *update* lokasi.



Gambar 3.2. *Flow Chart* Kondisi Idle

3.2.1.1. IMSI attach

Pada prosedur ini, MS meminta untuk mengalokasikan nomor IMSI ke sentral. Setelah sentral mengirimkan persetujuan untuk alokasi, baru kemudian MS mengirimkan nomor IMSI. Berikut algoritmanya. Listing program dapat dilihat pada Lampiran 1.

```

loop1   baca alamat IMSI
        kirim ke A
        kirim A ke P1
  
```

Respon yang diberikan sentral berupa nomor TMSI untuk di simpan pada memori kartu SIM. Pada program ini TMSI di simpan di memori internal alamat 30h – 3eh. Setelah menerima semua nomor TMSI, MS akan mengirimkan pemberitahuan bahwa alokasi TMSI selesai.

```
baca P1
  kirim P1 ke A
  jika A = 0
    kembali ke loop1
  hapus alamat TMSI
  simpan A ke alamat TMSI
  kirim pemberitahuan ke P1
```

3.2.1.2. Identifikasi ME

Identifikasi ME merupakan prosedur untuk mengecek keabsahan ME dengan melihat nomor IMEI. Prosedur ini diprakarsai oleh sentral dengan meminta nomor IMEI. Berikut algoritmanya, sedangkan listing program dapat dilihat pada Lampiran 2.

```
loop1  baca P1
        kirim P1 ke A
        jika A = 0
          kembali ke loop1
        kirim data alamat IMEI ke A
        kirim A ke P1
        hapus A

        baca P1
        kirim P1 ke A
        jika A = 0
          kembali ke loop1
        selesai
```

Dengan permintaan sentral tersebut, MS mengirimkan nomor IMEI yang tersimpan pada memori internal dengan alamat 40h – 4eh.

3.2.1.3. Meng-update Lokasi

MS akan meng-*update* lokasi ketika MS tersebut sudah berada pada area berbeda dari yang sebelumnya, artinya nilai LAI yang tersimpan pada memori kartu SIM harus di-*update*.

Selama MS berada pada kondisi *idle*, sentral akan terus berhubungan dengan MS, sehingga saat ada indikasi untuk melakukan *location update* sentral akan mengirimkan permintaan nomor LAI. Setelah itu sentral akan langsung mengirimkan nomor LAI dimana MS saat ini berada untuk

diperiksa oleh MS bilamana nomor LAI berubah sehingga membutuhkan *update* lokasi.

Di misalkan nilai LAI berubah sehingga MS perlu melakukan *update* lokasi, maka selanjutnya akan ada prosuder penyimpanan nomor LAI pada memori internal alamat 50h – 54h.

Berikut algoritma untuk prosedur *update* lokasi.

```
loop1   baca P1
        kirim P1 ke A
        jika A • data alamat LAI
        hapus A
loop2   kirim permintaan update ke A
        kirim A ke P1
        hapus A

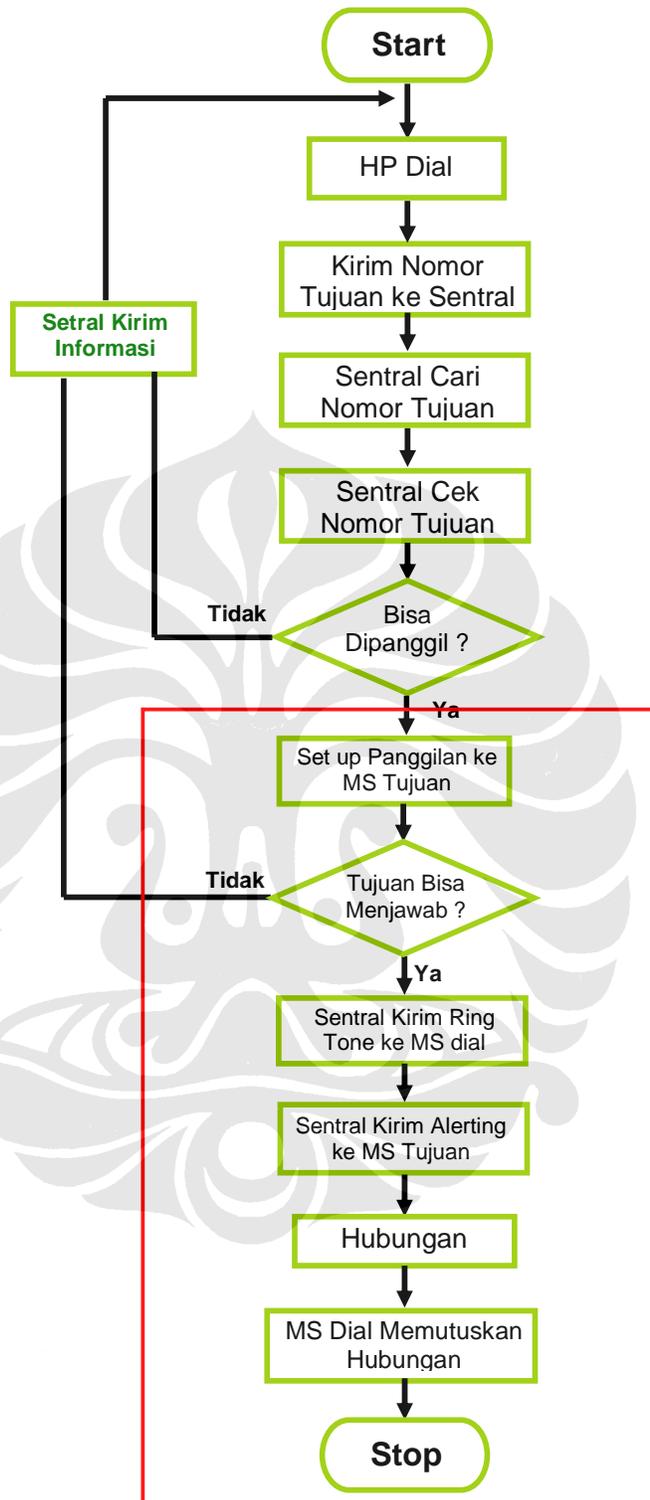
        baca P1
        kirim P1 ke A
        jika A = 0
        kembali ke loop1
        hapus data alamat LAI
        simpan A ke alamat LAI
        hapus A
```

Listing program dapat dilihat pada lampiran 3.

3.2.2. Panggilan Masuk

Flow chart pada Gambar 3.3 dibawah, menunjukkan pensinyalan saat proses panggilan. *Flow chart* yang diberi kotak dengan garis merah merupakan yang terjadi pada MS penerima. Yaitu pensinyalan antara sentral dengan MS penerima. *Flow chart* sebelumnya merupakan pensinyalan dari MS pengirim dengan sentral.

Ketika sentral mendapat pesan untuk menghubungkan MS pengirim dengan nomor yang di tuju, maka sentral tersebut akan memeriksa pada HLR-nya nomor tujuan dimana keberadaan MS tujuan tersebut. Setelah itu pesan *pagging* akan dikirim ke MSC yang melingkupi daerah tempat MS tujuan berada. MSC tersebut akan melakukan *pagging* ke semua BTS yang merupakan ruang lingkupnya dan kemudian akan diteruskan ke semua MS yang sedang di-*servicing* oleh BTS yang berada di ruang lingkup tersebut.



Gambar 3.3. Flow Chart Proses Panggilan Pada MS

MS yang memiliki nomor tersebut akan membalas pesan *pagging* tersebut dengan melakukan permintaan kanal. Berikut algoritmanya.

```
loop1    baca P1
         kirim P1 ke A
         jika A = 0
         kembali ke loop1
loop2    kirim permintaan pembangunan kanal ke P1
         hapus A
```

Kemudian sentral akan mengirimkan pesan tentang persetujuan penggunaan kanal. Namun ada kemungkinan sentral menolak permintaan kanal tersebut akibat kanal yang penuh. Apabila permintaan kanal ditolak maka MS akan kembali meminta kanal ke sentral. Berikut algoritmanya.

```
loop3    baca P1
         kirim P1 ke A
         jika A = 0
         kembali ke loop2
         hapus A

         baca P1
         kirim P1 ke A
         jika A = 0
         kembali ke loop3
```

Jika kanal tersedia untuk melakukan hubungan, maka sentral akan meminta permintaan panggilan ke MS. Kemudian MS akan mengirimkan pesan konfirmasi panggilan ke sentral, yang artinya MS penerima siap untuk menerima panggilan. Dengan siapnya MS penerima menerima panggilan, maka sentral akan mengizinkan hubungan antara MS pengirim dan MS penerima, jalur komunikasi akan terbentuk, bersamaan dengan di kirimnya permintaan *alerting* di MS penerima. Berikut algoritmanya.

```
kirim data alamat konfirmasi panggilan ke P1
;ring tone pada MS dial diaktifkan

baca P1
kirim P1 ke A
kirim A ke P0
hapus A
```

Pada rancang bangun sistem pensinyalan ini, *alerting* yang digunakan adalah menggunakan nyala LED. Apabila ditekan tombol 'yes' maka hubungan akan terjadi dan nyala LED. Sedangkan jika ditekan tombol 'no' maka MS akan mengirimkan pesan sibuk ke sentral yang kemudian akan diteruskan ke MS pengirim, lalu pensinyalan akan berakhir.

Apabila tombol 'no' yang ditekan, pesan sibuk akan dikirim, lalu hubungan akan berakhir. Namun, apabila yang ditekan tombol 'yes', hubungan akan terjadi serta MS akan selalu *stand by* untuk menerima pesan pengakhiran panggilan dari MS pengirim. Berikut algoritmanya.

```

    baca P2
    kirim P2 ke A
    jika A = 0
        jika waktu > 30 detik
            kirim perintah sibuk ke P1
        selesai

    jika A ≠ 0
        ; terjadi hubungan
        hapus A

```

Pesan persetujuan pengakhiran panggilan pun akan dikirim ke sentral untuk menandakan bahwa kanal siap dilepas. Algoritmanya dituliskan sebagai berikut.

```

    baca P2
    kirim P2 ke A
    jika A = n
        kirim A ke P1
        hapus A

    baca P1
    kirim P1 ke A
loop4  kirim laporan pengakhiran panggilan selesai ke A
        kirim A ke P1
        hapus A

    selesai

```

Listing program saat ada panggilan masuk sampai panggilan berakhir terdapat pada Lampiran 4.

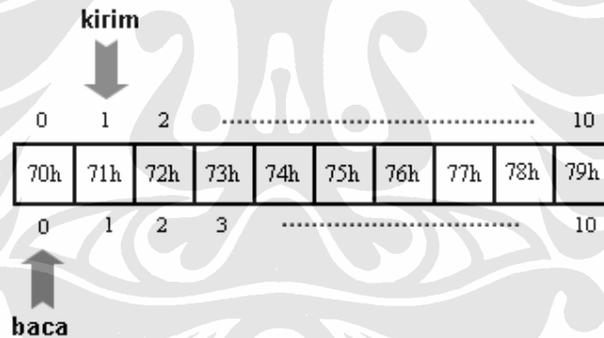
3.2.3. Delay

Untuk melihat keluaran pada port 1 yang menuju sentral maka pada programnya ditambahkan fungsi *delay*. Fungsi *delay* tersebut dipanggil setiap akan menunjukkan keluaran *microcontroller* pada port 1. Penulis membuat waktu *delay* selama 3 detik. Listing Program prosedur *delay* dapat dilihat pada Lampiran 5.

3.2.4. Sentral

Microcontroller ini juga diprogram sebagai sentral yang digunakan untuk memberikan balasan atas perintah ataupun data yang dikirim ke port 1 yang berfungsi sebagai sentral.

Pada memori internal *microcontroller* dibuat beberapa blok menyerupai suatu saluran untuk tempat mengalir data yang menuju sentral. Dalam skripsi ini, penulis menggunakan alamat memori internal 70h – 79h. Data yang di kirim ke sentral, masuk melewati saluran ini, menempati blok alamat yang kosong dengan bantuan pengecekan pada posisi kirim dan posisi baca. Jika tidak ada data yang masuk maka posisi kirim dan posisi baca berada pada angka yang sama. Saat data pertama masuk, data disimpan pada blok pertama, lalu posisi kirim bergeser pada posisi 1. Posisi baca berubah ketika sentral mengirimkan respon pada *microcontroller*. Untuk lebih jelasnya, listing program sentral dapat dilihat pada Lampiran 6. Gambar 3.4 menunjukkan saluran tempat mengalirnya data menuju sentral.



Gambar 3.4. Penggambaran Saluran Menuju Sentral

BAB IV

HASIL UJI COBA DAN ANALISIS PERANCANGAN SIMULASI SISTEM PENSINYALAN

4.1. HASIL UJI COBA

Pada bagian ini dilakukan pengujian program simulator sistem pensinyalan, baik menggunakan *software* simulator maupun menggunakan rangkaian yang disusun. Analisis dilakukan dengan menilai keluaran rangkaian, untuk dibandingkan dengan konsep perancangan rangkaian pada Bab III.

4.1.1. Software 8051 IDE

Pengujian pertama dilakukan menggunakan *software* 8051 IDE. Pengujian ini untuk mengetahui kesalahan yang terjadi pada program yang dibuat. Sebelum dijalankan, diperiksa berapa banyak *error* yang masih terjadi. Pada Gambar 4.1 ditunjukkan gambar hasil pemeriksaan *error*.



Gambar 4.1. Hasil Pemeriksaan *Error*

Program diawali dengan deklarasi alamat dan dilanjutkan dengan pengisian variabel yang ada serta beberapa register. Variabel tersebut harus dideklarasikan terlebih dahulu agar *microcontroller* dapat membacanya sebagai bagian dari program. Pada dasarnya, variabel tersebut hanya mengisi tempat kosong yang terdapat pada memori *microcontroller*. Kemudian tempat tersebut diberi label sesuai dengan nama variable yang digunakan. *Listing* Programnya terlampir pada Lampiran 7.

Terdapat beberapa data yang dimasukkan terlebih dahulu, karena diasumsikan bahwa data tersebut sudah disimpan pada kartu SIM (IMSI, TMSI,

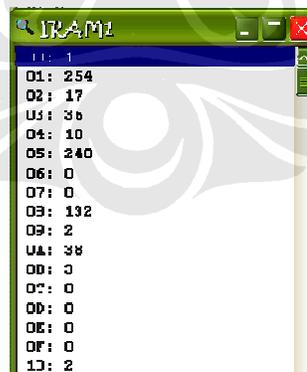
LAI, dan MCC) maupun pada MS (IMEI). *Listing* Programnya terdapat pada Lampiran 8. Nilai-nilai yang didefinisikan terlebih dahulu, terlihat pada Tabel 4.1 dibawah ini.

Tabel 4.1. Tabel Nilai Asumsi Sentral

Data	Alamat	Nilai Asumsi
IMSI	60h – 6eh	510 11 01 23456789
IMEI	40h 4eh	987654321987654
TMSI	30h – 3eh	123456789123456
LAI Lama	50h – 54h	51114
LAI Baru	50h – 54h	41587
Persetujuan Alokasi IMSI	02h	2
Persetujuan Nomor LAI	14h	7
Persetujuan Perbaruan Lokasi	15h	8
Persetujuan Permintaan Kanal	19h	1
Persetujuan <i>Service Request</i>	1bh	5
Permintaan <i>Alerting</i>	1dh	9
<i>Busy</i>	1eh	3
Pengakhiran Panggilan	23h	6

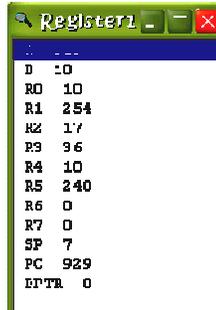
Pada nilai IMSI diatas, 510 merupakan nilai MCC, 11 merupakan nilai MNC yang merupakan penomoran jaringan di Indonesia, 01 merupakan identitas HLR, dan 8 nomor terakhir merupakan nomor pelanggan.

Tampilan memori internal pada simulator 8051 IDE ditunjukkan pada Gambar 4.2 berikut. Hasil yang diperlihatkan pada simulator ini sesuai dengan konsep perancangan yang ditentukan.



Gambar 4.2. Tampilan Memori Internal pada 8051 IDE

Program kemudian dijalankan sampai akhir, dan dilihat bagaimana keluarannya pada bagian register dan bagian port. Pada Gambar 4.3 berikut menampilkan isi register pada *microcontroller*.



Gambar 4.3. Tampilan Register pada 8051 IDE

Register 0 (R0) didefinisikan sebagai posisi kirim untuk data yang dikirim ke sentral. R4 sebagai posisi baca untuk mengirim respon dari sentral. Register 5, 6, dan 7 digunakan pada prosedur *delay*. Sedangkan register 1, 2, dan 3 digunakan sebagai register yang nilainya tidak tetap sebagai variabel. Hasil pengujian pada bagian register pun berjalan sesuai dengan konsep perancangan rangkaian ini.

Pada bagian port, hanya tiga port yang digunakan, yaitu port 0, port 1, dan port 3. Hasil keluaran akhir pada port ditunjukkan pada Gambar 4.4.



Gambar 4.4. Tampilan Port pada 8051 IDE

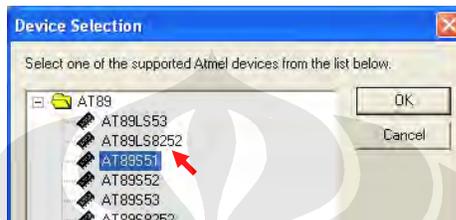
Secara keseluruhan, pada *software* 8051 IDE, program sistem pensinyalan ini berjalan sesuai fungsinya dan sesuai dengan konsep perancangan seperti yang dijelaskan pada Bab III, terutama keluaran pada portnya, yang dapat ditunjukkan melalui rangkaian simulator sistem pensinyalan.

4.1.2. Rangkaian Simulator Sistem Pensinyalan

Setelah dilakukan pengujian menggunakan *software* 8051IDE, maka pengujian selanjutnya adalah uji coba *hardware* rangkaian simulator sistem pensinyalan komunikasi seluler. Rangkaian simulator ini terdiri dari beberapa rangkaian terpisah, yaitu rangkaian sistem minimum *microcontroller* AT89S51, rangkaian port 0, dan rangkaian port 1.

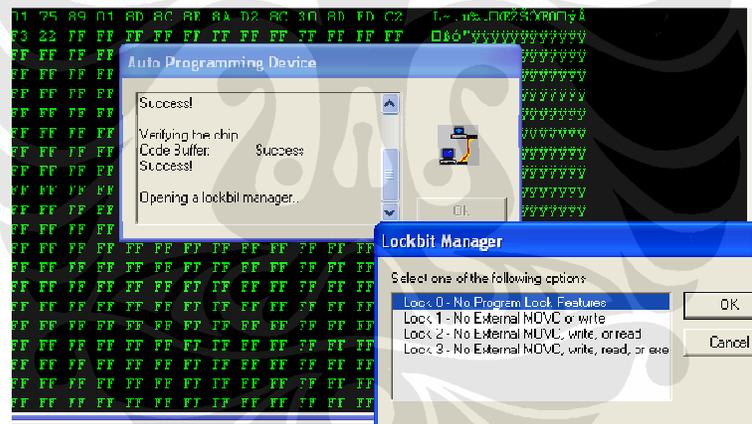
Pengujian *hardware* rangkaian simulator system pensinyalan komunikasi seluler bagian *subscriber* penerima dilakukan setelah program di-*burn* ke dalam *microcontroller*. Dengan menggunakan *software* ISP *microcontroller* Atmel seluruh program di-*burn* ke dalam *microcontroller*. Tahap-tahap yang dilakukan saat proses *burn*, yaitu :

1. Hubungkan sistem minimum ke komputer dengan kabel paralel,
2. Option → Device selection → Pilih AT89S51



Gambar 4.5. Tampilan Device Selection

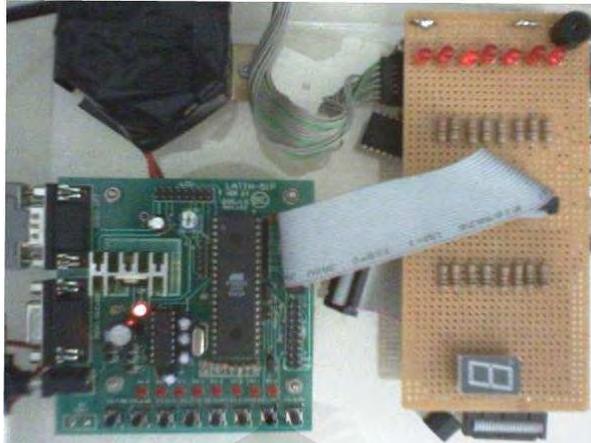
3. Load Buffer → Pilih program yang akan di-*burn* dalam bentuk *.hex*
4. Instruction → Auto program



Gambar 4.6. Tampilan saat Proses *Burning* Selesai

5. Instruction → Run Target

Setelah program di-*burn* kedalam *microcontroller*, bagian rangkaian yang ada disatukan, kemudian rangkaian simulator sistem pensinyalan tersebut siap untuk diuji cobakan. Pada Gambar 4.7 berikut ditunjukkan rangkaian simulator sistem pensinyalan secara keseluruhan.

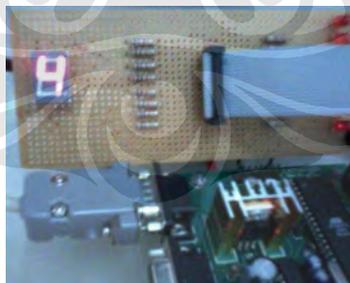


Gambar 4.7. Rangkaian Simulator Sistem Pensinyalan

4.1.2.1. Pengujian Port 0

Pengujian terlebih dahulu dilakukan dengan menguji rangkaian port 0. Rangkaian ini berfungsi sebagai penunjuk keluaran *microcontroller* yang menuju sentral.

Dengan program yang terdapat pada Lampiran 9 di dalam *microcontroller*, keluaran pada *seven segment* yang merupakan representasi dari saluran menuju sentral dapat ditunjukkan dan berjalan sesuai. Dengan munculnya angka 0,1,2,3, dan 4 menunjukkan bahwa rangkaian ini dapat berjalan sesuai dengan fungsinya. Rangkaian port 0 ditunjukkan pada Gambar 4.8 berikut.



Gambar 4.8. Rangkaian Port 0

Setelah dilakukan pengujian pada port ini, dicoba diujikan dengan program sistem pensinyalan yang dibuat. Hasil yang ditunjukkan juga sesuai dengan konsep perancangan sistem pensinyalan pada Bab III.

4.1.2.2. Pengujian Port 1

Rangkaian port 1 merupakan rangkaian LED. Rangkaian ini berfungsi sebagai peringatan panggilan masuk. Ketika ada panggilan masuk, LED akan menyala, dan apabila sudah ada perintah dari Port 3 maka LED akan mati.

Dengan program yang terdapat pada Lampiran 10, LED akan menyala dengan bergantian. Dengan berjalannya LED tersebut, maka rangkaian tersebut dapat berjalan sesuai fungsinya. Rangkaian pada port 1 ditunjukkan pada Gambar 4.9 berikut.



Gambar 4.9. Rangkaian Port 1

Hal yang sama juga dilakukan pada rangkaian port 1, rangkaian diujikan dengan program sistem pensinyalan yang dibuat. Hasilnya rangkaian berjalan sesuai dengan konsep perancangan sistem pensinyalan pada Bab III.

4.1.2.3. Pengujian Port 3

Port 3 merupakan rangkaian yang sudah terdapat pada sistem minimum yang digunakan. Rangkaian ini terdiri dari 8 *push button*. Tapi hanya dua yang digunakan, yaitu pin 0 dan pin 7.

Pengujian dilakukan dengan memasukkan program yang terdapat pada Lampiran 11. Saat *push button* pin 0 ditekan, lampu akan mati. Saat *push button* pin 1 ditekan, lampu akan menyala kembali. Pada Gambar 4.10 ditunjukkan rangkaian port 3.



Gambar 4.10. Rangkaian Port 3

Rangkaian port 3 ini juga dapat berjalan sesuai konsep perancangan yang ditentukan pada Bab III.

Setelah pengujian yang dilakukan pada tiap-tiap rangkaian berhasil, maka dilakukan uji coba terhadap sistem secara keseluruhan. *Microcontroller* di-burn dengan program pada Lampiran 1 sampai Lampiran 6 dan ketiga port dihubungkan ke rangkaian masing-masing.

Program kemudian dijalankan, dan berdasarkan hasil uji coba, hasilnya adalah rangkaian tersebut dapat berjalan sesuai dengan konsep perancangan pada Bab III. Secara umum jalannya simulasi bit-bit pensinyalan pada rangkaian simulator ini cukup baik. Hal tersebut dapat dilihat dari beberapa parameter sebagai berikut :

1. Semua bit-bit pensinyalan penting yang dikirimkan ke sentral dapat ditampilkan dengan lengkap dan baik.
2. Kedua tombol *push button*, baik yang berfungsi sebagai tombol 'yes' dan tombol 'no', dapat berjalan dengan baik. Lampu LED pada port 1, akan mati setelah kedua tombol tersebut ditekan.
3. Saat tombol 'no' ditekan, MS langsung mengakhiri panggilan yang ada, dan kembali ke kondisi *standby* untuk menerima panggilan. MS juga mengirimkan pemberitahuan pada sentral bahwa MS penerima sedang sibuk.
4. Saat tombol 'yes' ditekan, MS akan menyambungkan koneksi antara MS pengirim dan MS penerima. Lalu MS penerima akan berada dalam kondisi menunggu apabila ada perintah untuk mengakhiri panggilan atau MS penerima akan dapat mengakhiri panggilan ketika tombol 'no' ditekan.
5. Rangkaian port 0, sebagai penunjuk adanya panggilan dapat berjalan dengan baik, sesuai fungsinya. Lampu LED akan menyala saat perintah *alerting* dikirimkan dari sentral.

4.2. ANALISIS PERANCANGAN SISTEM PENSINYALAN

Perancangan rangkaian simulator sistem pensinyalan dapat dianalisa dari berbagai segi. Analisa dilakukan dengan membandingkan sistem pensinyalan yang sebenarnya, sehingga dapat diketahui kekurangan yang terdapat pada rangkaian sistem pensinyalan yang dirancang.

4.2.1. Permintaan

Berdasarkan studi pustaka, terdapat beberapa kekurangan dari simulator yang dirancang oleh penulis. Tidak semua permintaan yang dilakukan oleh MS dimasukkan ke dalam program yang dibuat dikarenakan keterbatasan memori pada *microcontroller* yang digunakan. Permintaan yang dimasukkan ke dalam program yang dibuat hanya permintaan untuk pensinyalan saja. Beberapa permintaan yang terdapat dalam program terdapat pada Tabel 3.1.

Selain karena keterbatasan memori *microcontroller*, penyesuaian ini dilakukan karena simulator sistem pensinyalan ini, memiliki bagian sentral yang juga disimpan dalam memori internal *microcontroller*. Sehingga semua data-data yang disimpan pada MS maupun sentral disimpan pada memori terlebih dahulu.

Permintaan yang dimasukkan di sini, dipilih yang paling penting untuk memperlihatkan aliran bit dari MS ke sentral serta dapat memodelkan sistem pensinyalan seperti aslinya.

4.2.2. Respon Sentral

Konsep perancang sistem pensinyalan ini idealnya memiliki dua *microcontroller* yang dihubungkan. Satu *microcontroller* berfungsi sebagai MS dan *microcontroller* lain berfungsi sebagai sentral. Namun karena pada pangaplikasiannya cukup sulit, maka hanya menggunakan satu *microcontroller* yang berfungsi utama sebagai MS. Namun didalamnya tetap diprogram program sentral untuk memberikan respon pada MS. Jadi respon sentral sebenarnya diberikan dari dalam *microcontroller* itu sendiri dengan program yang menyatu dengan program pada MS.

Selain penempatan sentral yang salah, yang disatukan dalam satu *microcontroller*, kekurangan sentral yang dibuat di sini adalah sentral yang dirancang terlalu sederhana. Dalam melakukan suatu respon, terkadang sentral

juga berkonsultasi dengan elemen jaringan lain seperti HLR, VLR, maupun EIR. Namun hal tersebut tidak menjadi pertimbangan perancangan, dikarenakan sentral tidak menjadi fokus dalam perancangan. Jadi, sentral di sini dibuat hanya untuk merespon keluaran dari *microcontroller* itu sendiri.

4.2.3. Penggunaan Tipe *Microcontroller*

Microcontroller yang digunakan pada rancang bangun kali ini adalah tipe Atmel 89S51 atau biasa disebut MCS-51. *Microcontroller* dengan arsitektur MCS-51 merupakan salah satu jenis arsitektur *microcontroller* yang paling lama dan paling banyak digunakan di dunia. Arsitektur ini dikeluarkan pertama kali oleh Intel dan kemudian menjadi sangat populer. Berbagai seri *microcontroller* berarsitektur MCS-51 telah diproduksi oleh berbagai vendor dan digunakan di dunia sebagai *microcontroller* yang bersifat *low cost, high performance*, dan mudah untuk didapatkan.

Selain karena alasan diatas, pemilihan tipe *microcontroller* AT89S51 dikarenakan *microcontroller* tipe AT89S ini sudah memiliki port ISP sehingga memudahkan untuk proses *burning*. Tabel 4.3 menunjukkan perbedaan beberapa tipe *microcontroller* Atmel.

Tabel 4.3. Tabel Perbedaan *Microcontroller* Atmel [13]

Seri	Flash (Kbytes)	RAM (Byte)	EEPROM (Kbytes)	Pin I/O	Timer	UART	SPI	ISP
AT89C51	4	128	-	32	2	1	-	-
AT89S51	4	128	-	32	2	1	-	Ya
AT89C52	8	256	-	32	3	1	-	-
AT89S52	8	256	-	32	3	1	-	Ya
AT89C2051	2	256	-	15	2	1	-	-
AT89S2051	2	256	-	15	2	1	-	Ya
AT89C4051	4	256	-	15	2	1	-	-
AT89S4051	4	256	-	15	2	1	-	Ya
AT89S8252	8	256	2	32	3	1	1	Ya
AT89S8253	12	256	2	32	3	1	1	Ya

4.2.4. Analisis Kejadian

Pemodelan sistem pensinyalan ini memiliki beberapa penyesuaian sehingga dapat dengan maksimal merepresentasikan sistem pensinyalan yang

sudah ada sekarang. Semua disesuaikan berdasarkan *microcontroller* yang dipilih dengan alasan yang sudah disebutkan pada bab II.

4.2.4.1. IMSI Attach

Pada saat IMSI *attach*, terdapat beberapa perbedaan sistem pensinyalan yang dirancang jika dibandingkan dengan sistem pensinyalan yang ada. Pertama, masalah frekuensi BTS. Pada sistem yang dirancang tidak dipertimbangkan masalah frekuensi tersebut. Ketika MS dinyalakan, MS melihat terhadap beberapa frekuensi BTS yang ada, dan kemudian melihat apakah MS yang digunakan dapat mendukung frekuensi tersebut. Jika tidak tentu MS harus mencari BTS lain untuk men-*servicing* MS tersebut. Kedua, masalah kuat sinyal yang ada. Saat MS dinyalakan, MS melihat kuat sinyal yang dimiliki setidaknya lima BTS terdekat, lalu BTS yang memiliki kuat sinyal paling besar, yang akan melayani MS tersebut. Selain itu, setelah dilakukan pemilihan terhadap BTS yang *servicing*, seharusnya MS selalu memantau kuat sinyal empat BTS tetangganya.

Namun, beberapa hal di atas disebabkan karena sistem pensinyalan yang dilakukan adalah sistem pensinyalan dari MS ke sentral, sesuai dengan batasan masalah yang sudah dijelaskan pada Bab I.

Pada IMSI *attach*, MS mengirimkan nomor IMSI ke sentral yang kemudian akan disimpan di HLR yang direspon oleh sentral dengan memberikan nilai TMSI. Namun karena tidak adanya memori khusus untuk sentral, nomor IMSI disimpan ke dalam memori internal *microcontroller* itu sendiri dan nomor TMSI sebenarnya sudah disimpan pada memori tersebut juga. Selain itu seharusnya saat sudah ditentukan BTS mana yang *servicing* MS tersebut, nomor LAI sudah disimpan di sentral. Namun pada perancangan, tidak ada prosedur tersebut, dikarenakan keterbatasan memori *microcontroller* yang ada.

4.2.4.2. Identifikasi IMEI

Identifikasi IMEI merupakan prosedur pengecekan keabsahan ME yang digunakan. Nomor IMEI yang dikirim ke sentral mestinya di periksa pada HLR. Namun pada perancangan kali ini, ME didefinisikan sah untuk

digunakan. Hal ini disebabkan, *microcontroller* hanya memiliki 4Kbyte EEPROM *non-volatile*.

4.2.4.3. Perbaruan Lokasi

Perbaruan lokasi harusnya dilakukan setelah melihat nomor LAI yang disimpan di kartu SIM berbeda dengan sel yang sedang *servicing*, karena pada kondisi *idle*, MS selalu melihat jika ada perubahan nomor LAI. Selain itu juga saat perbaruan lokasi, sentral mengirim lagi nomor TMSI yang baru. Namun karena keterbatasan yang dimiliki sistem minimum yang digunakan, hal tersebut diabaikan.

4.2.4.4. Menerima Panggilan

Setelah kondisi *idle*, sistem akan *stand by* untuk menerima panggilan ataupun melakukan panggilan. Sistem dirancang sebagai MS penerima yang menerima panggilan. Dilihat dari *message flow* yang terdapat pada Bab II, kondisi saat menerima panggilan adalah kondisi paling mendekati keadaan sebenarnya. Permintaan *paging* dari sentral, dibalas dengan permintaan kanal pensinyalan. Ketika sudah disetujui oleh sentral, sentral memulai untuk *set up* panggilan. Prosedur *set up* panggilan terakhir adalah memberikan permintaan untuk *alerting* MS.

Kekurangan yang terdapat pada sistem ini adalah nada sibuk. Pada program ini kita hanya mendefinisikan bahwa, apabila setelah 30 detik panggilan masuk tidak diangkat atau penerima menekan tombol 'no' maka, MS akan mengirimkan pesan sibuk ke sentral. Pesan sibuk ke sentral, akan diikuti dengan pemutusan panggilan. Sedangkan pada kondisi sebenarnya, pesan sibuk dapat diminta dari MS. Jadi, sentral dapat mengirimkan pesan sibuk yang berbeda ke MS pengirim.

Dari penjelasan diatas, sistem pensinyalan yang dirancang memiliki beberapa kekurangan, diantaranya :

1. Tidak adanya elemen jaringan lain seperti BTS, BSC, HLR, dan VLR sehingga jalannya proses pensinyalan tidak dapat sesuai dengan sistem pensinyalan sebenarnya.

2. Elemen MS dan MSC seharusnya memiliki program dan rangkaian terpisah, namun dalam rangkaian simulator ini disatukan. Hal tersebut menimbulkan kerancuan dalam proses serah terima bit-bit pensinyalan yang terjadi antara *mobile station* dan *mobile switching center*.
3. Akibat dari kekurangan pertama, prosedur perbaruan lokasi tidak termodelkan dengan baik. Hal tersebut disebabkan proses perbaruan lokasi membutuhkan setidaknya dua sentral sehingga aliran bitnya saat melakukan perbaruan lokasi dapat terlihat.
4. Pada pemilihan BTS yang *servicing*, tidak terjadi pemilihan BTS sesuai dengan kuat sinyal yang ada. Hal ini disebabkan elemen BTS yang ada diasumsikan hanya satu.
5. Perintah sibuk saat MS tidak dapat menerima panggilan hanya satu jenis. Sentral hanya dapat menanggapi respon sibuk dengan mengakhiri panggilan.

Sistem yang dirancang, belum dapat sepenuhnya termodelkan seperti sistem yang ada saat ini, namun sudah cukup baik untuk melihat aliran bit dari MS ke sentral mulai dari MS dinyalakan sampai MS dapat menerima panggilan. Semua penyesuaian yang dilakukan pada sistem yang dirancang tentu saja masih memperhatikan hal-hal penting sehingga sistem pensinyalan yang sebenarnya dapat termodelkan secara maksimal.

Kedepannya, untuk memodelkan sistem pensinyalan pada komunikasi seluler, diharapkan menggunakan *microcontroller* yang memiliki memori lebih besar, sehingga semua permintaan dapat dimodelkan semuanya. Selain itu karena arsitektur jaringan komunikasi seluler memiliki elemen yang beragam, diperlukan beberapa *microcontroller* untuk berfungsi elemen-elemen tersebut. Jadi, aliran bit saat MS pertama kali dinyalakan, saat MS melakukan panggilan dan saat MS menerima panggilan dapat dengan jelas di gambarkan.

BAB V

KESIMPULAN

Setelah melakukan perancangan rangkaian dan analisis terhadap rangkaian sistem pensinyalan yang berbasis *microcontroller* AT89S51, maka diperoleh beberapa kesimpulan, yaitu :

1. Hasil pengujian rangkain sistem pensinyalan yang dirancang, dapat berjalan sesuai dengan konsep perancangapn yang ditentukan.
2. Rangkaian sistem pensinyalan yang dirancang cukup untuk memodelkan sistem pensinyalan yang sebenarnya sesuai dengan kemampuan *microcontroller* yang digunakan, yaitu AT89S51.
3. Sistem ini memiliki kelemahan dalam proses aliran bit pensinyalan yang terjadi. Hal tersebut disebabkan karena program MS dan sentral yang terdapat pada satu *microcontroller*.
4. Sistem ini juga kurang dapat mengimplementasikan sistem pensinyalan sebenarnya akibat tidak adanya elemen jaringan lain seperti BTS, BSC, HLR, dan VLR.
5. Hasil perancangan ini masih dapat dikembangkan lebih lanjut dengan menambahkan beberapa *microcontroller* yang berfungsi sebagai elemen jaringan yang lain, seperti BTS, BSC, HLR, VLR, AuC, maupun EIR, sehingga sistem pensinyalan yang sebenarnya dapat termodelkan secara maksimal.

DAFTAR ACUAN

- [1] _____, “Teknologi Seluler,” Diakses 23 Juni 2008
http://www.e-dukasi.net/pengpop/pp_full.php?ppid=208&fname=semua.htm
- [2] _____, “Coomon Channel Signalling,” Bahan Kuliah Jaringan Telekomunikasi, Bandung : Program Studi Teknik Telekomunikasi JTE – STT TELKOM, 2006
- [3] _____, GSM System, White Paper : Nokia, 2005
- [4] Gunawan, Andi, “Sharing Knowledge,” Jakarta : PT. INDOSAT, 2007
- [5] Usman, Ule Kurniawan, “Sistem Komunikasi Bergerak,” Bandung : Lab SISKOM – STT Telkom
www.stt Telkom.ac.id/staf/UKU/Handout%20PT1123-DASTEL/Sistem%20Komunikasi%20Bergerak.ppt
- [6] _____, System Training : GSM Traffic Management, Training Document NOKIA, 2002
- [7] Goksel, Somer, ”Optimization and Log file Analysis in GSM,” 26 Januari 2003
- [8] GSM Communication Flow, Wireless Curriculum Development Section,
www.huawei.com
- [9] _____, “GSM Service,” Diakses 18 April 2008
http://en.wikipedia.org/wiki/GSM_services
- [10] _____, “GSM Signalling 2,” Jakarta : PT. INDOSAT
- [11] Data Sheet ATMEL 89S51
- [12] _____, “Non-volatile Memory,” Diakses 25 Mei 2008
http://en.wikipedia.org/wiki/Non-volatile_memory
- [13] _____, “Overview Microcontroller MCS-51,” Diakses 21 Juni 2008
<http://www.mikron123.com/content/view/48/53/>

LAMPIRAN

Lampiran 1 Listing Program Prosedur IMSI Attach

```
; Prosedur IMSI attach
mov p0,imsi_alloc
mov line_out,#00110010b      ;TAMPILIN 2
acall int
acall sentral
loop1: mov a,p0
      cjne a,imsi_alloc_app,loop1
      mov line_out,60h      ; kirim IMSI
      acall int
      mov line_out,61h
      acall int
      mov line_out,62h
      acall int
      mov line_out,63h
      acall int
      mov line_out,64h
      acall int
      mov line_out,65h
      acall int
      mov line_out,66h
      acall int
      mov line_out,67h
      acall int
      mov line_out,68h
      acall int
      mov line_out,69h
      acall int
      mov line_out,6ah
      acall int
      mov line_out,6bh
      acall int
      mov line_out,6ch
      acall int
      mov line_out,6dh
      acall int
      mov line_out,6eh
      acall int
                                     ;respon TMSI
      mov p0,#1
      mov 30h,p0
      mov p0,#2
      mov 31h,p0
      mov p0,#3
      mov 32h,p0
      mov p0,#4
      mov 33h,p0
      mov p0,#5
      mov 34h,p0
      mov p0,#6
      mov 35h,p0
      mov p0,#7
      mov 36h,p0
      mov p0,#8
      mov 37h,p0
      mov p0,#9
```

```

mov 38h,p0
mov p0,#1
mov 39h,p0
mov p0,#2
mov 3ah,p0
mov p0,#3
mov 3bh,p0
mov p0,#4
mov 3ch,p0
mov p0,#5
mov 3dh,p0
mov p0,#6
mov 3eh,p0
mov p0,akhir_tmsi
acall sentral

loop2:      mov a,p0
           cjne a,aloc_tmsi_fnsh,loop2

```

Lampiran 2 Listing Program Prosedur Identifikasi ME

```

; PROSEDUR Cek IMEI
mov p0,imei
acall sentral
ulg:      mov a,p0
           cjne a,iden_req,ulg

scd:      mov line_out,40h
           acall int
           mov line_out,41h
           acall int
           mov line_out,42h
           acall int
           mov line_out,43h
           acall int
           mov line_out,44h
           acall int
           mov line_out,45h
           acall int
           mov line_out,46h
           acall int
           mov line_out,47h
           acall int
           mov line_out,48h
           acall int
           mov line_out,49h
           acall int
           mov line_out,4ah
           acall int
           mov line_out,4bh
           acall int
           mov line_out,4ch
           acall int
           mov line_out,4dh
           acall int
           mov line_out,4eh
           acall int

```

Lampiran 3 Listing Program Prosedur Perbaruan Lokasi

```

; PROSEDUR Update Lokasi
mov p0,update_lok_req
mov line_out,#11010011b      ;TAMPILIN 7
acall int

```

```

cari:          acall sentral
              mov a,p0
              cjne a,LAI_req,cari
              mov p1,#4

;PROSEDUR CEK BERUBAH GAK NILAI LAI
satu:         mov a,p0                      ;baca dari port1
              cjne a,50h,update
dua:          mov a,p0
              cjne a,51h,update           ;cek bit kedua
tiga:         mov a,p0
              cjne a,52h,update           ;cek bit ketiga
empat:        mov a,p0
              cjne a,53h,update           ;cek bit keempat
lima:         mov a,p0
              cjne a,54h,update           ;cek bit kelima
update:       mov p0,up_lok_req
              mov line_out,#00010000b    ;TAMPILIN 8
              acall int

              acall sentral
              mov a,p0
              cjne a,LAI_aloc,update

; PROSEDUR SIMPEN LAI
mulai1:       mov line_out,#11010100b      ;4
              acall int
              mov a,p0                      ;baca dari sentral
              jnz simpen1
              sjmp mulai1
simpen1:      mov 50h,a
              mov line_out,#11010111b      ;1
              acall int
mulai2:       mov a,p0
              jnz simpen2
              sjmp mulai2
simpen2:      mov 51h,a
              mov line_out,#10011000b      ;5
              acall int
mulai3:       mov a,p0
              jnz simpen3
              sjmp mulai3
simpen3:      mov 52h,a
              mov line_out,#00010000b      ;8
              acall int
mulai4:       mov a,p0
              jnz simpen4
              sjmp mulai4
simpen4:      mov 53h,a
              mov line_out,#11010011b      ;7
              acall int
mulai5:       mov a,p0
              jnz simpen5
              sjmp mulai5
simpen5:      mov 54h,a

```

Lampiran 4 Listing Program Prosedur Panggilan Masuk

```

; PROSEDUR Telpon Masuk
standby:     clr a
pagging:     mov p0,siapasuk
              acall sentral
              mov a,p0
              jz pagging
              cjne a,pag_req,pagging

```

```

next:          mov line_out,#11010111b      ;TAMPILIN 1
              acall int
ats:          acall sentral

; PROSEDUR Set Up panggilan
kanal:       mov a,p0
              jz kanal
              cjne a,kanal_req_app,rjct
              sjmp trs

rjct:        cjne a,kanal_req_rjt,ats
              sjmp next

trs:         mov p0,service_req
              mov line_out,#10011000b      ;TAMPILIN 5
              acall int
              acall sentral

set:         mov a,p0
              jz set
              cjne a,service_req_app,pagging
              mov line_out,#10010000b      ;TAMPILIN 9
              acall int

delay:       acall sentral
              mov a,p0
              jz delay
              cjne a,alert_req,set

LED:         mov a,#11111110b
              mov r5,#240
loop3:       mov p1,#0
              rr a
              mov p2,a
              mov r2,#20h
              mov r3,#20h
delay1:      mov r1,p3
              cjne r1,#11111110b,no
              ajmp hubungan
delay2:      djnz r2,delay1
              djnz r3,delay1
              djnz r5,loop3
              ajmp sibuk

no:          cjne r1,#01111111b,delay2

sibuk:       mov line_out,busy
              acall int
              ajmp end

hubungan:    mov p2,#11111111b
              acall sentral
              mov a,p0
              cjne a,call_release,hubungan
              mov p0,call_rls_ok
              mov line_out,#00011100b      ;TAMPILIN 6
              acall int
              ajmp end

```

Lampiran 5 Listing Program Prosedur Delay

```
;Prosedur INT
```

```

int:          mov a,line_out
             mov p0,a

             mov r7,#20      ;20*50ms=3000ms=1s
t1ms:        mov r5,#76
             mov r6,#01
             mov tmod,#1    ;timer 0 in 16-bit mode
loops:       mov th0,r5     ;move R0 value to TH0
             mov tl0,r6     ;move R1 value to TL0
             setb tr0       ;make timer 0 start counting
             jnb tf0,$      ;wait until tf0 = 1
             clr tf0        ;clear flag tf0
             djnz r7,loops  ;wait until R2 = 0
             ret

```

Lampiran 6 Listing Program Sentral

```

; Prosedur Sentral
;r0 posisi kirim
;r4 posisi baca
sentral:     inc r0
             mov b,r0
             mov a,r4

;Prosedur Kirim
             cjne a,b,alamat1
             ret
alamat1:    mov r3,sal1
             cjne r3,#0,alamat2
             mov r2,p0
             mov sal1,r2
             acall baca
             ret
alamat2:    mov r3,sal2
             cjne r3,#0,alamat3
             mov r2,p0
             mov sal2,r2
             acall baca
             ret
alamat3:    mov r3,sal3
             cjne r3,#0,alamat4
             mov r2,p0
             mov sal3,r2
             acall baca
             ret
alamat4:    mov r3,sal4
             cjne r3,#0,alamat5
             mov r2,p0
             mov sal4,r2
             acall baca
             ret
alamat5:    mov r3,sal5
             cjne r3,#0,alamat6
             mov r2,p0
             mov sal5,r2
             acall baca
             ret
alamat6:    mov r3,sal6
             cjne r3,#0,alamat7
             mov r2,p0
             mov sal6,r2
             acall baca
             ret
alamat7:    mov r3,sal7
             cjne r3,#0,alamat8

```

```

                                mov r2,p0
                                mov sal7,r2
                                acall baca
                                ret
alamat8:                        mov r3,sal8
                                cjne r3,#0,alamat9
                                mov r2,p0
                                mov sal8,r2
                                acall baca
                                ret
alamat9:                        mov r3,sal9
                                cjne r3,#0,alamat10
                                mov r2,p0
                                mov sal9,r2
                                acall baca
                                ret
alamat10:                       mov r2,p0
                                mov sal10,r2
                                acall baca10
                                ret
;Prosedur baca
baca:                            cjne a,b,cek1
                                ret
cek1:                            cjne r0,#0,cek2
                                ret
cek2:                            cjne r0,#1,cek3
                                ajmp bacal
                                ret
cek3:                            cjne r0,#2,cek4
                                ajmp baca2
                                ret
cek4:                            cjne r0,#3,cek5
                                ajmp baca3
                                ret
cek5:                            cjne r0,#4,cek6
                                ajmp baca4
                                ret
cek6:                            cjne r0,#5,cek7
                                ajmp baca5
                                ret
cek7:                            cjne r0,#6,cek8
                                ajmp baca6
                                ret
cek8:                            cjne r0,#7,cek9
                                ajmp baca7
                                ret
cek9:                            cjne r0,#8,cek10
                                ajmp baca8
                                ret
cek10:                           cjne r0,#9,cek11
                                ajmp baca9
                                ret
cek11:                           cjne r0,#10,cek12
                                ajmp bacal0
                                ret
cek12:                           ret
bacal:                            inc r4
                                mov r1,imsi_aloc_app
                                mov p0,r1
                                ret
baca2:                            inc r4
                                mov r1,aloc_tmsi_fnsh
                                mov p0,r1
                                ret
baca3:                            inc r4
                                mov p0,iden_req
                                ret

```

```

baca4:      inc r4
            mov p0,LAI_req
            ret
baca5:      inc r4
            mov p0,LAI_aloc
            ret
baca6:      inc r4
            mov p0,pag_req
            ret
baca7:      inc r4
            mov p0,kanal_req_app
            ret
baca8:      inc r4
            mov p0,service_req_app
            ret
baca9:      inc r4
            mov p0>alert_req
            ret
baca10:     inc r4
            mov p0,call_release
            ret

```

Lampiran 7 Listing Program Deklarasi Variabel dan Deklarasi Alamat

```

;Deklarasi alamat
line_out    equ    02h
imsi_aloc   equ    10h
imsi_aloc_app equ    11h
aloc_tmsi_fnsh equ    12h
iden_req    equ    13h
LAI_req     equ    14h
up_lok_req  equ    15h
LAI_aloc    equ    16h
pag_req     equ    17h
kanal_req   equ    18h
kanal_req_app equ    19h
kanal_req_rjt equ    1ah
service_req_app equ    1bh
konf_call   equ    1ch
alert_req   equ    1dh
busy        equ    1eh
tilpun      equ    21h
terima      equ    22h
call_release equ    23h
call_rls_ok equ    24h
update_lok_req equ    25h
akhir_tmsi  equ    26h
imei        equ    27h
siapmasuk   equ    28h
service_req  equ    29h
sal1        equ    70h
sal2        equ    71h
sal3        equ    72h
sal4        equ    73h
sal5        equ    74h
sal6        equ    75h
sal7        equ    76h
sal8        equ    77h
sal9        equ    78h
sal10       equ    79h

;Deklarasi variabel
mov imsi_aloc,#00000010b
mov imsi_aloc_app,#11110000b

```

```

mov alloc_tmsi_fnsh,#01001011b
mov iden_req,#00011000b
mov LAI_req,#00011001b
mov up_lok_req,#00001000b
mov LAI_alloc,#00000100b
mov pag_req,#10000001b
mov kanal_req,#00000001b
mov kanal_req_app,#00000011b
mov kanal_req_rjt,#00001111b
mov service_req_app,#00100100b
mov konf_call,#00001001b
mov alert_req,#11111111b
mov busy,#11010111b ;#10101010b
mov tilpun,#01011010b
mov terima,#10100101b
mov call_release,#11011011b
mov call_rls_ok,#00000110b
mov update_lok_req,#00000111b
mov akhir_tmsi,#00110011b
mov imei,#00000101b ;#11100111b
mov siapmasuk,#11111110b
mov service_req,#00000101b
mov sal1,#0
mov sal2,#0
mov sal3,#0
mov sal4,#0
mov sal5,#0
mov sal6,#0
mov sal7,#0
mov sal8,#0
mov sal9,#0
mov sal10,#0
mov p0,#00000000b
mov r0,#0
mov r1,#0
mov r4,#0
mov r3,#0

```

Lampiran 8 Listing Program Nilai Pada Sentral

```

;Nilai-nilai pada Sentral

mov 40h,#10010000b ; IMEI
mov 41h,#00010000b
mov 42h,#11010011b
mov 43h,#00011100b
mov 44h,#10011000b
mov 45h,#11010100b
mov 46h,#10010010b
mov 47h,#00110010b
mov 48h,#11010111b
mov 49h,#10010000b
mov 4ah,#00010000b
mov 4bh,#11010011b
mov 4ch,#00011100b
mov 4dh,#10011000b
mov 4eh,#11010100b

mov 60h,#10011000b ; IMSI
mov 61h,#11010111b
mov 62h,#00010001b
mov 63h,#11010111b
mov 64h,#11010111b
mov 65h,#00010001b
mov 66h,#11010111b

```

```

mov 67h,#00110010b
mov 68h,#10010010b
mov 69h,#11010100b
mov 6ah,#10011000b
mov 6bh,#00011100b
mov 6ch,#11010011b
mov 6dh,#00010000b
mov 6eh,#10010000b

mov 50h,#5           ; LAI
mov 51h,#1
mov 52h,#1
mov 53h,#1
mov 54h,#4

```

Lampiran 9 Listing Program Tes Port 0

```

line_out      equ      00h

; Prosedur Tes Port 0
mov line_out,#00010001b
acall tampilkan
mov line_out,#11010111b
acall tampilkan
mov line_out,#00110010b
acall tampilkan
mov line_out,#10010010b
acall tampilkan
mov line_out,#11010100b
acall tampilkan

tampilkan:    mov a,line_out
              mov p0,a

              mov r7,#20      ;20*50ms=3000ms=1s
tlms:        mov r5,#76
              mov r6,#01
              mov tmod,#1    ;timer 0 in 16-bit mode
loops:       mov th0,r5      ;move R0 value to TH0
              mov tl0,r6      ;move R1 value to TL0
              setb tr0        ;make timer 0 start counting
              jnb tf0,$       ;wait until tf0 = 1
              clr tf0        ;clear flag tf0
              djnz r7,loops   ;wait until R2 = 0
              ret

```

Lampiran 10 Listing Program Tes Port 1

```

LED:         mov a,#11111110b
loop3:       rr a
              mov p2,a
              mov r2,#20h
              mov r3,#20h
delay1:      djnz r2,delay1
              djnz r3,delay1
              acall loop3

```

Lampiran 11 Listing Program Tes Port 3

```

awal:

```

```
mov r0,p3
cjne r0,#11111110b,mati
mov p2,#00000000b
mati:
cjne r0,#11111101b,awal
mov p2,#11111111b
```

