UNIVERSITAS INDONESIA

RANCANG BANGUN SIMULASI ENKRIPSI PADA KOMUNIKASI GSM

SKRIPSI

PERMADI HUDOYO JUNRAMDLAN
06 06 04 285 3

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2008

UNIVERSITAS INDONESIA

RANCANG BANGUN SIMULASI ENKRIPSI PADA KOMUNIKASI GSM

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

PERMADI HUDOYO JUNRAMDLAN
06 06 04 285 3

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2008

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

Nama : Permadi H Junramdlan

NPM : 06 06 04 285 3

Tanda Tangan : ……

Tanggal : 22 Desember 2008

HALAMAN PENGESAHAN

Tugas akhir ini diajukan oleh :

Nama	: Permadi Hudoyo Junramdlan
NPM	: 06 06 04 285 3
Program Studi	: Teknik Elektro
Judul Tugas akhir	: **RANCANG BANGUN SIMULASI ENKRIPSI PADA KOMUNIKASI GSM**

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing	: Dr. Ir. Arman Djohan Diponegoro, M. Eng	(……)
Penguji	: Fitri Yuli Zulkifli, ST, M.Sc	(…….)
Penguji	: Dr.Ir. Agus Santoso Tamsir, MT	(.....)

Ditetapkan di : Universitas Indonesia Depok

Tanggal : 22 Desember 2008

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Allah Yang Maha Esa, karena atas kasih sayang dan rahmat-Nya, penulis dapat menyelesaikan tugas akhir ini. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Dalam pembuatan tugas akhir ini, bukanlah semata-mata karena usaha dan kerja individu penulis sendiri, tetapi mendapatkan banyak bantuan dari berbagai pihak, untuk itu penulis menyampaikan terima kasih kepada:

(1) Allah SWT, Yang Maha Kuasa atas seluruh kehidupan.
(2) Bapak Dr. Ir. Arman Djohan Diponegoro, M.Eng, selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi, dan bimbingan serta persetujuan sehingga tugas akhir ini dapat selesai dengan baik.
(3) Ibu Dr. Fitri Yuli Zulkifli, ST, M.Sc dan Bapak Dr.Ir. Agus Santoso Tamsir, MT.
(4) Teman –teman yang tanpa mengurangi rasa hormat, untuk tidak disebutkan.

Akhir kata, saya berharap Allah Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini bermanfaat bagi pengembangan ilmu.

Depok, 22 Desember 2008

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama	: Permadi Hudoyo Junramdlan
NPM	: 06 06 04 285 3
Program Studi	: Teknik Elektro
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis Karya	: Tugas akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

RANCANG BANGUN SIMULASI ENKRIPSI PADA KOMUNIKASI GSM

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 22 Desember 2008

Yang Menyatakan,

(Permadi Hudoyo Junramdlan)

ABSTRAK

Nama : Permadi Hudoyo Junramdlan

Program Studi : Teknik Elektro

Judul : **RANCANG BANGUN SIMULASI ENKRIPSI PADA KOMUNIKASI GSM**

Simulasi ini diawali dengan melakukan pembangkitan *cipering key* sebagai syarat untuk melakukan enkripsi data informasi. Untuk pembangkitan *ciphering key* pada komunikasi GSM, digunakanlah algoritma A8 yang akan melakukan seluruh komputasi data Ki dan RAND yang dibutuhkan. Setelah itu, *ciphering key* yang telah diperoleh akan diproses oleh algoritma A5 dengan tujuan mengacak informasi.

Kata Kunci :

Algoritma A8, Algoritma A5, *ciphering key*, enkripsi, dekripsi, Ki, dan RAND.

ABSTRACT

Name : Permadi Hudoyo Junramdlan

Study Program : Electrical Engineering

Title ***:Design and Construction of Encryption Simulation on GSM Communication.***

This simulation start of to generated of ciphering key for data encryption. To generated this key base on GSM communication, used the A8 algorithm and to get data encryption used of A5 algorithm.

Key words:

Algorithm of A8, Chipering Key, Kc, Ki, dan RAND.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

DAFTAR SINGKATAN

LCD	*Liquid Crystal Display*
SIM	*Subscriber Indentity Module*
ME	*Mobile Equipment*
BTS	*Base Transceiver Station*
BSC	*Base Station Controller*
HLR	*Home Location Register*
VLR	*Visitor Location Register*
MSC	*Mobile Service Switching center*
EIR	*Equipment Identity Register*
AuC	*Authentication Centre*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era perkembangan teknologi sekarang ini, kejahatan dalam sektor telekomunikasi yang lebih terkenal dengan istilah *fraud* merupakan hal yang merugikan baik bagi *customer* maupun operator. *Fraud* tidak hanya terjadi pada sistem komunikasi tetap namun juga pada komunikasi bergerak. *Fraud* yang banyak terjadi dalam komunikasi bergerak adalah penggandaan identitas dan nomor telepon selular tertentu atau mencoba menyusup kepada jaringan telepon bergerak padahal tidak terdaftar sebagai customer pada operator telepon selular tersebut. [1]

Fraud merupakan hal yang serius bagi operator komunikasi bergerak. Jumlah kerugian akibat *fraud* rata-rata mencapai 500 juta dollar setiap tahun dalam industri komunikasi bergerak di Amerika, yang merupakan suatu angka yang cukup besar untuk membangun jaringan sistem komunikasi baru. Banyak kejadian *fraud* pada komunikasi telepon gengam seperti pada kasus di Arizona dimana seorang melakukan 57 ribu kali panggilan interlokal maupun internasional selama 19 hari dengan menggunakan identitas palsu sehingga operator menderita kerugian sebesar sekitar 1 juta dollar. Akhirnya dapat ditangkap melalui kerjasama antara operator dan polisi setempat berikut sekitar 10 ribu *microchip* palsu termasuk komputer dan software yang dipergunakan untuk memprogram kembali chip pada telepon gengam lain. Cara yang dapat dilakukan untuk mencegah dan mendeteksi *fraud* diantaranya adalah dengan melakukan *encryption*.

1. 2 Perumusan Masalah

Adapun perumusan masalah pada sistem yang akan dibuat ini adalah bagaimana proses pembangkitan Kc (*chipering key)* dengan komputasi data menggunakan algoritma A8.

1. 3 Tujuan

Tujuan dari tugas akhir ini adalah mensimulasikan proses pembangkit Kc (*chipering key)* menggunakan algoritma *A8* melalui mikrokontroller dan diuji coba dalam proses enkripsi dan dekripsi.

1. 4 Batasan Masalah

Adapun batasan masalah pada tugas akhir ini, adalah:

a) *Software* yang diinginkan adalah aljabar algoritma A8.
b) Nilai *input Ki* dan *RAND* sebesar masing-masing 128 bit *fixed* dilakukan secara manual melalui *keypad* dan ditampilkan melalui *LCD* 2 x 16 karakter.

1. 5 Metodologi

Untuk menyelesaikan makalah ini, dilakukan langkah-langkah sebagai berikut:

a) Mempelajari konsep tentang pengamanan GSM pada telekomunikasi.
b) Dilakukan proses aljabar algoritma A8 menggunakan mikrokontroller.
c) Rancang bangun suatu implementasi algoritma A8.
d) Dilakukan proses uji coba Kc untuk enkripsi dan dekripsi dengan program berbasis PC.

1. 6 Sistematika Pembahasan

Pada tugas akhir ini terdiri dari 5 (lima) bab, dimana masing-masing bab mempunyai kaitan satu sama lain, yaitu:

BAB I PENDAHULUAN

Memberikan latar belakang tentang permasalahan, tujuan, masalah dan batasan masalah yang dibahas dalam tugas akhir ini.

BAB II LANDASAN TEORI

Memberikan tinjauan pustaka yang berkaitan dengan algoritma A8 yang digunakan untuk pembangkit Kc (*chipering key)*. Membahas teori dasar yang menunjang RANCANG BANGUN sistem termasuk diantaranya dasar-dasar mikrokontroler *AT89S52*.

BAB III RANCANG BANGUN SISTEM

Membahas RANCANG BANGUN sistem yang dibuat baik *hardware* maupun *software*. Antara lain mengenai pembuatan sistem kontrol peralatan, *set-up* pada *port* mikrokontroler, serta *software* komputasi algoritma *A8*.

BAB IV PENGUJIAN DAN ANALISA

Berisi data akhir output serta analisa mengenai proses terjadinya dengan melakukan simulasi melalui *keypad* terhadap mikrokontroller.

BAB V PENUTUP

Berisi kesimpulan dari dasar-dasar sistem dan RANCANG BANGUN sistem.

BAB 2
TINJAUAN PUSTAKA

2. 1 Mikrokontroller ATMEL AT89S52[2]

Mikrokontroler AT89S52 merupakan mikrokontroler CMOS 8-bit yang mempunyai tegangan rendah dimana memiliki kemampuan yang tinggi dengan 8 Kbyte *Flash Programmable* and *Erasable Read Only Memory* (PEROM) atau lebih dikenal dengan *In System Programmable Flash Memory*. Piranti ini memiliki teknologi memori *non volatile* dengan kerapatan tinggi dari Atmel yang kompatibel dengan mikrokontroler standar industri MCS-51 baik pin kaki IC maupun set instruksinya.

AT89S52 ini memiliki *on-chip flash* yang memberikan memori program untuk dapat diprogram ulang kembali ke dalam system yang dilakukan oleh *programmer*. Kombinasi sebuah *versatile* CPU 8-bit dengan menanamkan *flash* memori di dalamnya menjadi sebuah keping monolitik (*monolithic chip*). AT89S52 juga menyediakan cukup banyak instruksi sehingga teknik pemrogramannya sangat mudah yang memungkinkan pembuat program dapat menggunakan dengan fleksibel dengan harga yang murah dan aplikasi-aplikasi yang banyak diperoleh. Selain itu mikrokontroler AT89S52 juga memilii beberapa fitur lainnya, seperti:

a) Kompatible dengan keluarga mikrokontroler MCS-51.
b) 8 Kbyte *In-system Programmable* (ISP) *flash* memori sehingga memiliki kemampuan dapat diprogram sampai 1000 kali pemprograman (baca/tulis).
c) Tegangan kerja 4.0 – 5.5 V.
d) Bekerja pada frekuensi 0 – 33 MHz.
e) Tiga level program *memory lock*.
f) 256 x 8 bit RAM internal.
g) 32 jalur I/O yang dapat diprogram.
h) Tiga buah *Timer/ Counter* 16 Bit.
i) Delapan sumber *interrupt*.
j) Saluran UART serial *Full Duplex*.

k) *Watchdog Timer*.

l) *Mode low-power idle* dan *Power-down*.

m) *Interrupt recovery* dari modul *power-down*.

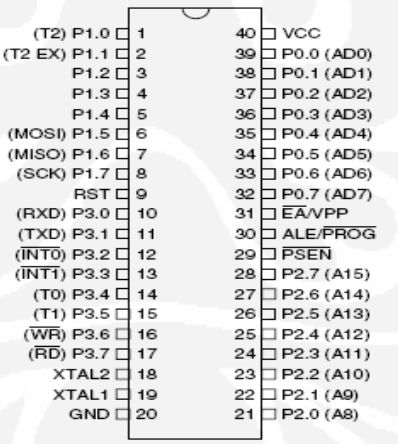
n) *Dual data pointer*.

o) Mode pemprograman ISP yang fleksible (*Byte* dan *Page Mode*).

AT89S52 dirancang dengan logika statis untuk operasi hingga frekuensi nol dan mendukung penyimpangan daya dua buah perangkat lunak (*software*) untuk pemilihan mode operasi. Mode *idle* menghentikan CPU dan membiarkan RAM, *timer/counter*, port serial, dan sistem interupsi untuk terus berfungsi. Mode *power-down* menyimpan isi RAM tetapi membekukan osilator, menon-aktifkan seluruh fungsi *chip* sampai ada interupsi eksternal atau reset pada *hardware*.

2. 1. 1 Konfigurasi Pin AT89S52

Pada Gambar 2.1 di bawah ini, AT89S52 mempunyai 40 kaki, 32 kaki digunakan untuk keperluan port parallel. Dimana setiap port terdiri atas 8 pin, sehingga terdapat 4 port, yaitu: port 0, port 1, port 2, port 3.



[2]

Gambar 2. 1 Bentuk Konfigurasi pin AT89S52.

Dari Gambar 2.1 di atas, berikut ini merupakan spesifikasi dari port – port paralel yang mikrokontroler AT89S52, yaitu:

a) Port 0

Port 0 merupakan port I/O 8 bit jalur bidirectional terbuka, yang berfungsi sebagai port output dan pada masing-masing pin dapat memasukkan 8 input TTL. Pada saat '1' dituliskan ke pin port 0, sehingga pin ini dapat berfungsi sebagai input impedansi tinggi. Port 0 dapat juga dikonfigurasikan pada multiplexed low order address/ data bus selama akses ke program eksternal dan memori data, dan pada mode ini P0 mempunyai pull-up internal. Port 0 juga menerima kode byte selama Flash programming dan mengeluarkan kode byte selama program verifikasi.

b) Port 1

Port 1 adalah port I/O 8 bit bidirectional dengan pull-up internal. Port 1 output buffer dapat menjadi sumber 4 TTL input. Ketika '1' ditulis ke pin port 1, pin di-pull high oleh pull-up internal dan dapat digunakan sebagai input. Sebagai input, pin port 1 yang secara eksternal di-pull low akan menjadi sumber arus (I_{IL}) karena berasal dari pull-up internal. Port 1 juga menerima low-order address byte selama Flash programming dan verification. Berikut ini adalah fungsi lain dari Port 1, yaitu:

a. P1.0 : eksternal input counter terhadap timer / counter 2, clock out (T2).
b. P1.1 : Timer/counter 2 capture/reload trigger/direction control (T2EX).
c. P1.5 : MOSI (Digunakan untuk in system programming).
d. P1.6 : MISO (Digunakan untuk in system programming).
e. P1.7 : SCLK (Digunakan untuk in system programming).

c) Port 2

Port 2 adalah port I/O 8 bit bidirectional dengan pull-up internal. Output buffer port 2 dapat menjadi 4 sumber TTL input. Ketika '1' ditulis ke pin port 2, pin dapat di pull high oleh pull-up internal dan dapat digunakan sebagai input. Dimana jika sebagai input, pin port 2 yang secara eksternal di pull-low akan menjadi arus sumber (I_{IL}) karena berasal dari pull up internal. Port 2 mengeluarkan *high-order address byte* selama pengambilan dari memori program eksternal dan selama akses ke memori data eksternal menggunakan 16 bit alamat (MOVX@DPTR). Dalam aplikasinya menggunakan internal *pull-up* yang kuat ketika mengeluarkan '1'. Selama akses ke memori data eksternal mengeluarkan alamat 8 bit (MOVX@R1), port 2 mengeluarkan isi port 2 Special

Function Register. Port 2 juga menerima high order address bit dan beberapa sinyal control selama Flash programming dan verification.

d) Port 3

Port 3 adalah port I/O 8 bit bidirectional dengan *pull-up internal*. Keluaran buffer port 3 dapat menjadi sumber 4 TTL input. Ketika '1' dituliskan ke port 3, pin di pull-high oleh internal *pull-up* dan dapat digunakan sebagai input. Sebagai input, pin port 3 yang di-pull low sumber arus (I_{IL}) karena adanya pull-up internal. Serta menerima pula beberapa sinyal control untuk Flash Programming dan verification. Port ini juga mempunyai fungsi lain, yaitu:

1. P3.0 : RXD (Serial Input Port).
2. P3.1 : TXD (Serial Output Port).
3. P3.2 : 1NT0 (Eksternal Interrupt 0).
4. P3.3 : 1NT1 (Esternal Interrupt 1).
5. P3.4 : T0 (Timer 0 Eksternal Input).
6. P3.5 : T1 (Timer 1 Eksternal Input).
7. P3.6 : WR (Eksternal Data Memori Write Strobe).
8. P3.7 : RD (Eksternal Data Memori Read Strobe).

Mikrokontroler AT89S52 selain memiliki port – port parallel, piranti ini juga di lengkapi dengan perangkat komunikasi serial. Untuk mengaktifkan dan mengkonfigurasinya, *programmer* harus mengakses register SCON dan bit SMOD (bit ke-7 pada register PCON). Dimana perangkat komunikasi serial pada mikrokontroler AT89S52 dapat dioperasikan dalam 4 mode, yaitu:

a. Mode 0

Merupakan sarana komunikasi data seri sinkron, data seri dikirim dan diterima melalui kaki RxD, sedangkan kaki TxD dapat dipakai untuk menyalurkan clock yang diperlukan komunikasi data sinkron. Data ditransmisikan per 8 bit dengan kecepatan transmisi data (Baud rate) tetap sebesar $^1/_2$ frekuensi kerja AT89S52.

b. Mode 1

Mode 1 dan dua mode berikutnya merupakan sarana komunikasi seri asinkron. Data seri dikirim melalui kaki TxD dan diterima dari kaki RxD. Data

ditransmisikan per 10 bit yang terdiri atas 1 bit start (‘0’), 8 bit data, dan 1 bit stop (‘1’). Kecepatan transmisi data (baud rate) ditentukan lewat timer 1 yang bisa diatur untuk berbagai kecepatan.

c. Mode 2

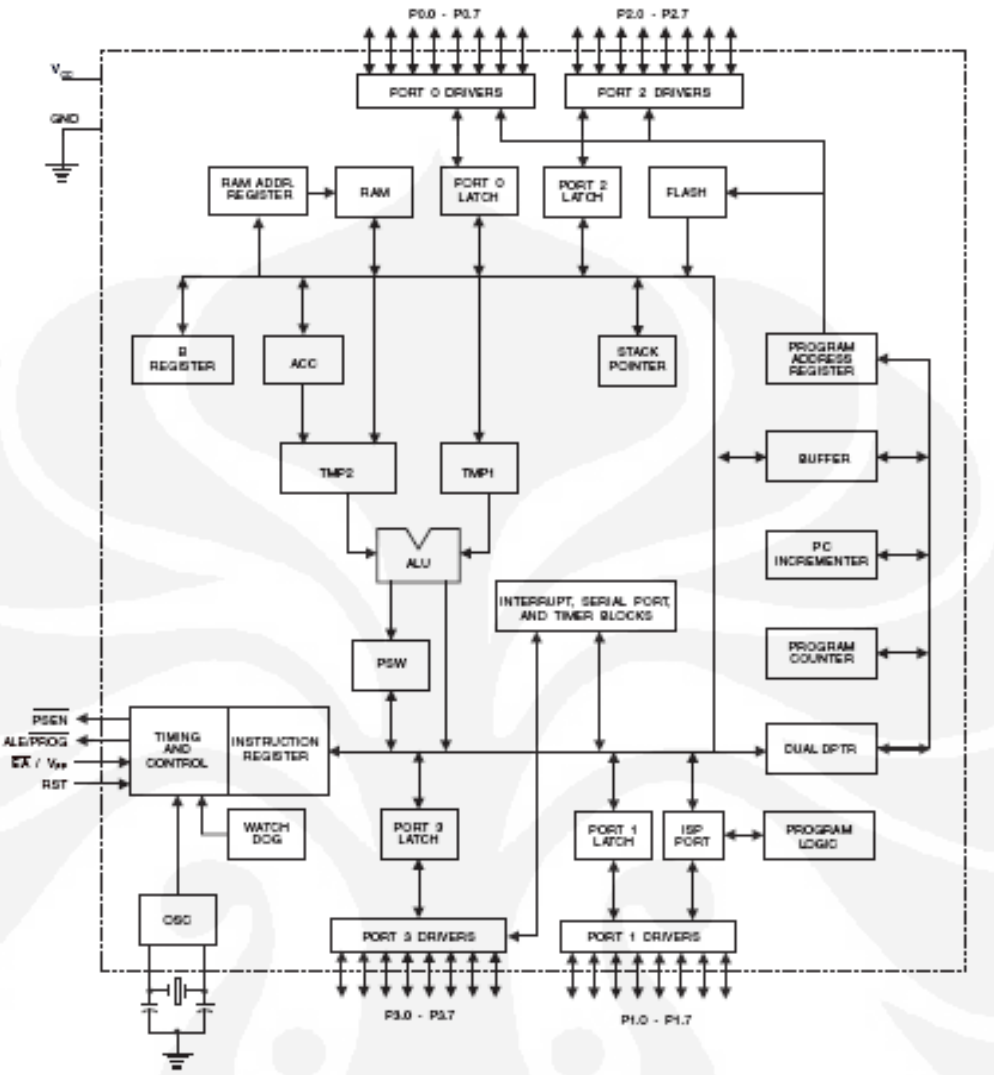
Data seri dikirim melalui kaki TxD dan diterima dari kaki RxD. Data ditransmisikan per 11 bit, terdiri atas 1 bit start (‘0’), 8 bit data, 1 bit data tambahan (bit ke-9), dan 1 bit stop (‘1’). Kecepatan transmisi data (baud rate) hanya dapat dipilih $^1/_{32}$ atau $^1/_{64}$ frekuensi kerja AT89S52.

d. Mode 3

Data seri dikirim melalui kaki TxD dan diterima dari kaki RxD. Data ditransmisikan per 11 bit juga. Pada dasarnnya mode 2 dan mode 3 sama persis. Perbedaannya adalah kecepatan transmisi data (baud rate) mode 3 ditentukan lewat timer 1, yang bisa diatur untuk berbagai kecepatan, persis sama dengan mode 1.

2. 1. 2 Arsitektur Mikrokontroler AT89S52

Pada Gambar 2. 2 di bawah ini, memperlihatkan arsitektur mikrokontroler AT89S52.



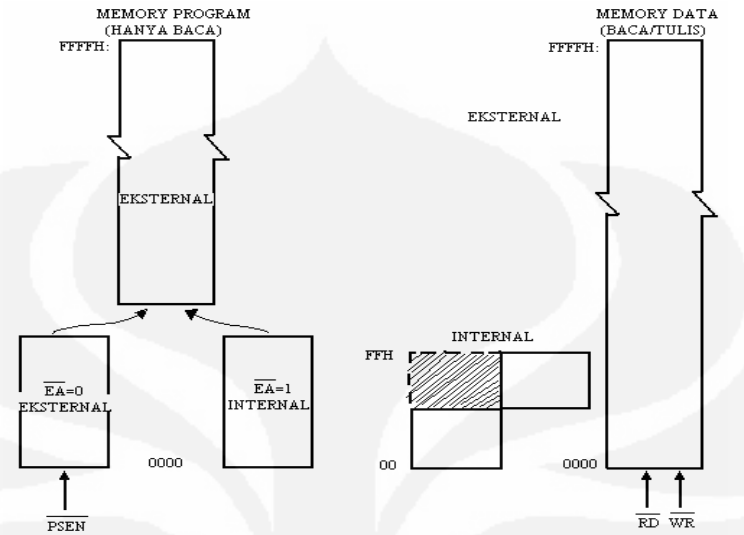
[2]

Gambar 2. 2 Arsitektur AT89S52.

2. 1. 3 Organisasi Memory

Semua perangkat MCS-51, termasuk AT89S52, memiliki ruang alamat memori data dan program yang terpisah. Dimana Program memori dikhususkan untuk menyimpan program, hanya bisa dibaca, sedangkan data memori untuk menyimpan data-data yang bisa berubah dalam proses, bisa baca dan tulis.

Dimana pada Gambar 2. 3 memperlihatkan struktur memory dan data pada AT89S52.



[3]

Gambar 2. 3 Struktur memori program dan data pada AT89S52.

Dari Gambar 2.3 tersebut di atas, Pemisahan memori program dan data tersebut membolehkan memori data diakses dengan alamat 8 bit, sehingga dapat dengan cepat dan mudah disimpan dan dimanipulasi oleh CPU 8 bit. Namun demikian, alamat memori data 16 bit bisa juga dihasilkan melalui register DPTR (Data pointer).

a. Memori Program

   Memori program hanya bisa dibaca saja. Terdapat memori program yang bisa diakses langsung hingga 64K byte. Sedangka strobe untuk akses program memori eksternal melalui sinyal Program Store Enable.

b. Memori Data

   Memori data menempati suatu ruang alamat yang terpisah dari memori program. Memori eksternal dapat diakses secara langsung hingga 64K byte dalam ruang memori data eksternal. CPU akan memberikan sinyal baca dan tulis, selama pengaksesan memori data eksternal.

c. Flash PEROM

   Untuk menyimpan program secara permanen, AT89S52 menyediakan *Flash* PEROM dengan kapasitas 4 Kbyte, yaitu suatu ROM yang dapat ditulis ulang atau dihapus menggunakan *programmer.*

d. SFR (Special Function Register)

Mikrokontroler mempunyai peta memori yang dikenal sebagai Special Function Register (SFR). SFR pada mikrokontroler dibagi menjadi beberapa bagian serta mempunyai alamat masing-masing, seperti pada Gambar 2.4.

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

[2]

Gambar 2.4 Peta SFR dan nilai resetnya.

Pada Gambar 2. 4 tersebut di atas terlihat bagian sisi kiri dan kanan dituliskan alamat-alamatnya dalam format heksadesimal. Tidak semua alamat pada SFR digunakan dan diimplementasikan pada chip. Jika dilakukan pembacaan pada alamat yang tidak terpakai tersebut akan menghasilkan data acak dan penulisannya tidak menimbulkan effek sama sekali.

Berikut ini adalah beberapa SFR dan alamatnya:

a) Accumulator : Menyimpan data sementara (E0H).
b) Register B : Operasi perkalian dan pembagian (F0H).
c) Program Status word (PSW) : Informasi Status Program (D0H).
d) Stack Pointer : Menyimpan dan mengambil data dari atau ke stack (81H).

e) Data Pointer : Menampung data 16 bit (83H dan 82H).Port 0, 1, 2, 3 : Menyimpan data yang akan dibaca atau ditulis dari atau ke port (80H, 90H, A0H).
f) Serial Data Buffer : Sebagai register penyangga penerima atau pengirim (99H).
g) Timer Register : Merupakan register-register pencacah 16 bit untuk masing-masing timer 0, 1, dan 2.
h) Capture Register : Menyimpan nilai isi ulang (CBH dan CAH).

e. Mode-mode pengalamatan

a) Pengalamatan langsung (Direct Addressing)
b) Dalam pengalamatan langsung, pemindahan data ditentukan berdasarkan alamat 8 bit (1 byte) dalam suatu instruksi. Hanya RAM data internal dan SFR yang dapat diakses secara langsung.
c) Pengalamatan tak langsung (Indirect Addressing)
d) Dalam pengalamatan tak-langsung, instruksi menentukan suatu register yang digunakan untuk menyimpan alamat operand. Baik RAM internal maupun eksternal dapat diakses secara tak-langsung. Register alamat untuk alamat-alamat 8 bit bisa menggunakan stack pointer atau R0 atau R1 dari bank register yang dipilih. Sebaliknya, alamat 16 bit hanya bisa menggunakan register pointer data 16 bit atau DPTR.
e) Pengalamatan Terindeks (Indexed Addressing)
f) Memori program hanya bisa diakses melalui pengalamatan terindeks. Mode pengalamatan ini ditujukan untuk membaca label look-up (look-up tables) yang tersimpan dalam memori program. Sebuah register dasar 16 bit menunjuk ke awal atau dasar tabel dan akumulator di-set dengan angka indeks tabel yang dapat diakses. Alamat dari entri tabel dalam memori program dibentuk dengan menjumlahkan data akumulator dengan penunjuk awal tabel.

2. 2 DT-51 MINIMUM SYSTEM.[4]

DT-51 adalah alat pengembangan mikrokontroller keluarga MCS-51TM yang sederhana, handal, dan ekonomis. DT-51 berbentuk sistem minimum dengan komponen utamanya mikrokontroller AT89S52. DT-51 memungkinkan dalam mengembangkan aplikasi digital dengan mudah; menulis software (perangkat lunak) pada komputer yang kemudian men-download ke board DT-51, dan menjalankannya; serta dapat langsung bekerja sendiri (stand-alone) pada sistem yang ada tanpa penggantian / penambahan komponen.

Minimum Sistem mikrokontroler merupakan sebuah kit mikrokontroler yang sudah dapat berfungsi sebagai pengontrol utama suatu sistem elektronika. Kit DT-51 merupakan kit yang lengkap untuk dapat digunakan sebagai board utama karena telah tersedia port serial, input data, memori eksternal 28C64B, dan 1 buah PPI 8255. DT-51 juga telah dilengkapi dengan driver dan port LCD yang memudahkan kita bila ingin menghubungkan LCD ke board. Spesifikasi DT-51 sebagai berikut :

a. Berbasis mikrokontroler AT89S52 yang berstandar industri.
b. Serial port interface standar RS-232 untuk komunikasi antara komputer dengan board DT-51.
c. 8 Kbytes non-volatile memory (EEPROM) untuk menyimpan program dan data.
d. 4 por t input output (I/O) dengan kapasitas 8 bit tiap por tnya.
e. Port *Liquid Crystal Display* (LCD) untuk keperluan tampilan.
f. Konektor ekspansi untuk menghubungkan DT-51 dengan add-on board yang kompatibel dari Innovative Electronics.

2. 2. 1 Peta Memori DT-51

Peta Memori DT-51 menunjukkan alamat masing-masing bagian komponen sebagai berikut :

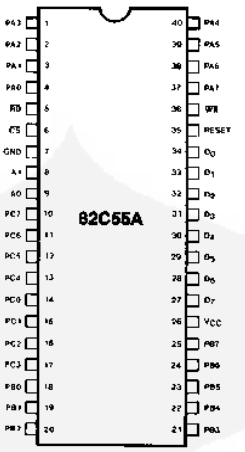
a. 0000H - 1FFFH, 8 Kbyte pertama digunakan sebagai internal dan 4 Kbyte PEROM yang berisi kernel code, sedangkan 4K sisanya reserved.
b. 2000H - 3FFFH, 8 Kbyte kedua digunakan untuk PPI 8255 dan hanya terpakai 4 alamat :
c. 2000H - Port A
d. 2001H - Port B
e. 2002H - Port C
f. 2003H - Control Word Register
g. 4000H - 5FFFH, 8 Kbyte ketiga digunakan oleh EEPROM untuk menyimpan User Code.
h. 6000H – FFFFH, CS3-CS7 disediakan untuk ekspansi.

Pada memori internal DT-51 sudah diisi dengan kernel yang tidak bisa ditulis ulang kembali. Oleh karena itu, DT-51 menggunakan memori eksternal AT28C64B, yaitu Electrically Erasable and Programmable Read Only Memory (EEPROM) kualitas tinggi berukuran 64 KByte, yang terdiri dari 8.192 words berukuran 8 bit, sehingga memiliki ukuran program yang lebih besar.

2. 2. 2 PPI 82C55 (*Programmable Peripheral Interface*).[5]

Programmable Peripheral Interface (PPI) 8255 adalah komponen antarmuka yang sangat populer serta murah dan merupakan chip antarmuka 24 bit (3 port) yang dapat diprogram kerjanya sesuai keinginan. PPI 8255 merupakan chip yang paling banyak digunakan untuk interfacing computer yang dihubungkan ke port ISA computer.

Di bawah ini adalah Gambar 2.5 IC PPI 82C55 sebagai komponen antarmuka.



[5]

Gambar 2. 5 Pin-Out dari adapter antarmuka peripheral (PPI) 8255.

Pada Gambar 2.5 tersebut di atas, yang merupakan pin kaki IC 8255 yang terdiri dari 40 pin, dimana pin Gnd berada pada pin 7 dan Vcc pada pin ke 26. Berikut ini merupakan detugas akhir dari masing-masing pin:

a. PA0 – PA7, Pin ini merupakan port A yang terdiri dari 8 bit yang dapat diprogram sebagai input atau output dengan mode bidirectional input/output.
b. PB0 – PB7, Port B ini dapat diprogram sebagai input/output tetapi tidak dapat digunakan sebagai port bidirectional.
c. PC0 – PC7, Port C ini dapat diprogram sebagai input/output bahkan dapat dipecah menjadi 2, yaitu CU (bit PC4 – PC7) dan CL (bit PC0 – PC3).
d. RD dan WR, Sinyal kontrol aktif rendah ini dihubungkan ke 8255. Jika 8255 menggunakan desain peripheral I/O, IOR, dan IOW dari sistem bus, maka akan dihubungkan ke kedua pin ini.
e. RESET, Pin aktif tinggi ini digunakan untuk membersihkan (clear) control register. Ketika RESET diaktifkan, seluruh port akan diinisialisasi sebagai port input.

Untuk IC 82C55 dipilih dari pin Control select (CS) untuk pemprograman dan untuk membaca atau menulis ke suatu port. Pemilihan register dilaksanakan melalui pin–pin masukan A0 dan A1 yang memilih suatu register internal untuk pemprograman

atau operasi. Dimana pada Table 2.1, menunjukkan tugas port I/O yang dipakai untuk memprogram dan mengakses port I/O.

Tabel 2. 1. Pemilihan port I/O untuk 8255.

CS	A1	A0	Fungsi
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Register
1	X	X	8255 tidak dipilih

[5] ”telah diolah kembali”

Pada saat port A, B, dan C digunakan sebagai I/O, maka mode operasi port tersebut perlu di-set. Ada empat mode operasi yang dimiliki 8255, yaitu:

a. Mode 0 (Basic input / output). Merupakan mode yang paling sederhana, dimana semua port dapat diprogram sebagai input/output. Pada mode ini seluruh port sebagai output atau input dan tidak ada port ynag dapat dikontrol secara individual.
b. Mode 1 (Strobe input / output). Pada mode ini port A dan B dapat digunakan sebagai input atau output dengan kemampuan *handshaking*. Sinyal *handshaking* disediakan oleh bit-bit port C.
c. Mode 2 (*Bidirectional bus*). Port A dapat digunakan sebagai port *bidirectional* I/O dengan kemampuan *handshaking*, dimana sinyalnya disediakan oleh port C. Port B dapat digunakan sebagai model I/O sederhana atau mode 1 *handshaking*.
d. Mode BSR (Bit *Set* / *Reset*). Dengan mode ini, hanya port individual port C saja yang dapat diprogram.

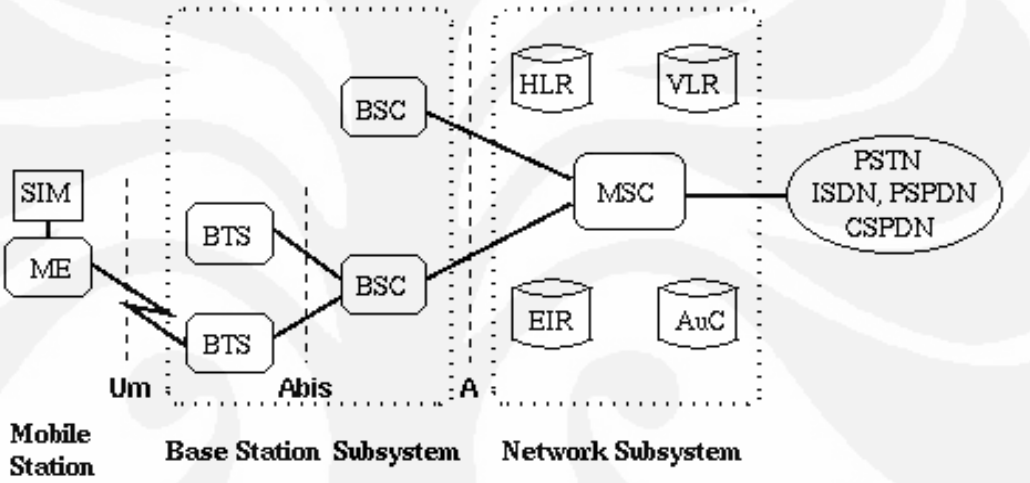
2.3. Global System for Mobile Communication [7]

Global Sistem for Mobile communication (GSM) adalah sebuah standar global untuk komunikasi bergerak digital. GSM adalah nama dari sebuah group

standarisasi yang dibentuk di Eropa tahun 1982 untuk menciptakan sebuah standar bersama telpon bergerak selular di Eropa yang beroperasi pada daerah frekuensi 900 MHz. GSM saat ini banyak digunakan di negara-negara di dunia. Jaringan GSM dibagi tiga bagian yaitu:

a. *The base subscriber carries the mobile station. (MS)*
b. *The base station subsystem.*
c. *The network subsystem.*

Gambar 2.6 *Lay-out* keseluruhan arsitektur jaringan dari sistem *global system for mobile communication* (GSM).



Gambar 2.6 *Layout generic* dari jaringan GSM menurut John's Scourias [3]

Keterangan Gambar 2.6:

- (ME)/HP = Mobile Equipment	- SIM =Subscriber Identification Module
- BTS = Base Transceiver Station	- BSC = Base Station Controller
- MSC = Mobile Service Switching Centre	- HLR = Home Location Register
- VLR = Visitors Locations Register	- EIR = Equipment Identity Register
- AuC = Authentication Center	- PSTN = Public Switched Telepone network
- ISDN =Integrated Service Digital Network	

2.3.1 *Mobile Station* (MS)

MS merupakan perangkat *outstation* yang biasanya berupa pesawat telepon, yang digunakan pelanggan untuk mendapatkan pelayanan dari jaringan GSM tertera pada Gambar 2.7yang terdiri dari:

Gambar 2.7 Bagian dari pembentuk *Mobile Station* (MS) [4]

2.3.2 *Mobile Equipment* (ME)

ME adalah perangkat yang berupa pesawat teleponnya, serta berfungsi sebagai:

- Unit kontrol, yaitu peralatan di mana pelanggan dapat memilih nomor-nomor yang akan dikirim,
- Unit *transceiver*, yang berfungsi menghubungkan MS dengan BSS melalui hubungan radio dua arah.

Setiap ME memiliki *International Mobile Equipment Identity* (IMEI) yang digunakan untuk mencegah penggunaan perangkat ME yang mengalami suatu pencurian.

a) *Subscriber Identity Module* (SIM) *card.*

SIM card merupakan smart card di mana di dalamnya terdapat microprosessor (umumnya 8 bit), RAM, ROM dan EPROM, sehingga selain menyimpan data dapat pula melakukan proses komputasi [5]. Jika kita melihat dimensi fisiknya, maka dikenal 2 macam SIM card, yaitu IC card SIM (memenuhi standard ISO-7816) yang memiliki ukuran sebesar kartu kredit dan *plug-in* SIM yang memiliki ukuran 15 x 25 mm.

SIM card berfungsi sebagai Unit Logic, yaitu merupakan pengontrol utama terhadap peralatan MS dan digunakan untuk menyimpan informasi untuk mendukung operasi dan pelayanan sistem GSM yang berhubungan dengan proses autentikasi pelanggan. SIM card berisi nomor khusus dari pelanggan yang disebut *International Mobile Subscriber Identity* (IMSI).

Untuk keperluan keamanan, SIM card juga melakukan mekanisme pengecekan keabsahan antara pemakai dengan MS dengan menggunakan PIN (*Personal Identification Number)* atau CHV (*Card Holder Verification).* Demi alasan keamanan, PIN dapat diganti oleh pemakai jika PIN yang lama sudah diketahui oleh orang lain.

Pemakai harus memasukkan serangkaian digit (biasanya 4-8 digit) untuk dapat memakai perangkat MS-nya. SIM card akan membandingkan digit tersebut dengan data PIN yang tersimpan dalam SIM. Jika sama maka pengaksesan dapat dilanjutkan, sedangakan jika berbeda akan diberi kesempatan tiga kali mencoba dan jika tidak sama juga, maka SIM card akan diblok. Pembukaan *blocking* dapat dilakukan oleh pemakai dengan menggunakan fasilitas PUK (*Personal Unblocking Key).* Pembukaan *blocking* dibatasi sampai sepuluh kali dan setelah itu SIM card tidak dapat dipakai lagi sehingga harus diganti dengan yang baru.

Sebagai tambahan untuk keperluan pengecekan otoritasi pelanggan tersebut, maka SIM card harus menyediakan kapabilitas penyimpanan informasi tentang PIN (*Personal Identity Number*), *indicator enable/disable* PIN, peghitungan kesalahan PIN, PUK (*Personal Unblocking Key*), data autentikasi.

2.3.3 Arsitektur Jaringan GSM [6]

Suatu jaringan GSM tersusun atas beberapa fungsionalitas, yang fungsinya maupun antarmukanya telah dispesifikasikan. Jaringan GSM dapat dibagi menjadi 3 bagian besar. *Mobile Station* (MS) dibawa oleh pelanggan. *Base Station Subsystem* (BS) mengendalikan jalur radio dengan MS. *Network Subsystem*, yang mempunyai bagian

utama disebut *Mobile Services Switching Center* (MSC), melaksanakan pensaklaran panggilan antar pengguna bergerak dengan pengguna jaringan tetap. MSC juga menangani operasi pengelolaan mobilitas. Selain itu ada *Operations and Maintenance Center*, yang menjamin operasi yang benar dan setup jaringan. MS dan BSS berkomunikasi melalui antarmuka, yang dikenal sebagai antarmuka udara atau jalur radio. BSS berkomunikasi dengan MSC melalui antarmuka A. Beberapa istilah berikut akan dibahas :

SIM = *Subscriber Indentity Module*
ME = *Mobile Equipment*
BTS = *Base Transceiver Station*
BSC = *Base Station Controller*
HLR = *Home Location Register*
VLR = *Visitor Location Register*
MSC = *Mobile Service Switching center*
EIR = *Equipment Identity Register*
AuC = *Authentication Centre*

2.3.4 Stasiun Bergerak (MS)

MS tersusun atas peralatan bergerak (terminal) dan suatu *smart card* yang disebut *Subscriber Identity Module* (SIM). SIM menyediakan mobilitas personal, sehingga pengguna dapat mengakses layanan yang ia langgan tidak tergantung terminalnya. Dengan menyisipkan kartu SIM kedalam terminal lain, pengguna dapat menerima panggilan dari terminal tersebut, membuat panggilan dari terminal tersebut, dan menerima layanan lainnya.

Peralatan bergerak secara unik diidentifikasi dengan *International Mobile Equipment Identity* (IMEI). Kartu SIM berisi *International Mobile Subscriber Identity* (IMSI) yang dipakai untuk mengidentifikasi pelanggan sistem, sebuah kunci rahasia untuk autentikasi, dan informasi lainnya. IMEI dan IMSI adalah independen, sehingga

memungkinkan mobilitas personal. Kartu SIM dapat diproteksi terhadap pemakaian yang tidak sah dengan sebuah *password* atau PIN (*Personal Identification Number*).

2.3.5 Subsistem Base Station

Subsistem base station tersusun atas dua bagian, *Base Transceiver Station* (BTS) dan *Base Station Controller* (BSC). Keduanya berkomunikasi menggunakan antar muka standar Abis, yang memungkinkan operasi antara komponen-komponen yang dibuat oleh pabrik yang berbeda (interoperabilitas).

BTS mempunyai *tranceiver* 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 persyaratan: tahan banting (*ruggedness*), reliabilitas, portabilitas, dan biaya minimum.
BSC mengelola sumberdaya radio untuk satu atau lebih BTS. Ia menangani setiap saluran radio, *frequency hopping*, dan *handover*. BSC merupakan koneksi antara stasiun bergerak dengan MSC.

2.3.6 Subsistem Jaringan

Komponen utama subsistem jaringan adalah MSC. Ia berperanan seperti halnya *switching node* pada PSTN atau ISDN, dan selain itu menyediakan semua fungsionalitas yang diperlukan untuk menangani pelanggan bergerak, seperti halnya registrasi, autentikasi, *location updating, handover*, dan *call routing* ke pelanggan yang *roaming*. Layanan-layanan ini disediakan berkaitan dengan berbagai entitas fungsional, yang bersama-sama membentuk subsistem jaringan. MSC menyediakan koneksi ke jaringan tetap (PSTN atau ISDN). Pensinyalan antara entitas fungsional pada subsistem jaringan menggunakan *Signalling System Number* 7 (SS7), yang dipakai untuk *trunk signaling* pada ISDN dan secara meluas dipakai pada jaringan publik.

HLR dan VLR bersama dengan MSC, menyediakan kompatibilitas *call-routing* dan *roaming* dari GSM. HLR berisi semua informasi administratif dari setiap pelanggan

yang tercatat pada jaringan GSM yang berkaitan, bersama-sama dengan lokasi saat itu dari unit bergerak. Lokasi unit bergerak biasanya dalam bentuk alamat pensinyalan dari VLR yang berkaitan dengan stasiun bergerak. Secara logika ada sebuah HLR untuk tiap jaringan GSM, sekalipun dapat diimplementasikan sebagai basis data terdistribusi.

VLR berisi informasi administratif terpilih dari HLR, yang diperlukan untuk pengendalian panggilan dan penyediaan layanan-layanan, untuk tiap unit bergerak yang saat ini berlokasi di daerah yang secara geografis dikendalikan oleh VLR. Sekalipun tiap entitas fungsional dapat diimplementasikan sebagai suatu unit independen, semua manufakturer piranti *switching* hingga saat ini masih mengimplementasikan VLR bersama dengan MSC, sehingga area geografis yang dikendalikan oleh MSC sama dengan area yang dikendalikan VLR, sehingga menyederhanakan persyaratan pensinyalan. Harap dicatat bahwa MSC tidak berisi informasi tentang stasiun bergerak tertentu. Informasi ini disimpan pada register lokasi**.**

2.3.7 *Subscriber Identity Module (SIM)*

SIM menyediakan identitas ke ME. SIM tidak lain adalah sebuah *smart card* yang memiliki CPU dan memori. Parameter pelanggan disimpan di SIM. SIM mempunyai EEPROM dan ROM. ROM berisi algoritma A3 dan A8. EEPROM berisi IMSI dan Ki. PIN memproteksi SIM terhadap penggunaan yang tidak sah. PUK (**Personal Unblocking Key**) memproteksi terhadap pemasukan PIN yang salah berikutnya, terlihat pada Tabel 2.2 tentang ukuran memori pada SIM *card.*

Tabel 2.2. Memori pada SIM card

Memori	**Ukuran Umum**	**Ukuran Maksimum**
ROM	4-6 Kbyte	16 Kbyte
RAM	126-160 Byte	256 Byte
EEPROM	2-3 KByte	8 KByte

2.3.8 *Home Location Register (HLR)*

HLR menyimpan identitas dan data pengguna dari semua pelanggan di area tersebut. Ini meliputi IMSI, Ki, ijin layanan suplementer, dan beberapa data temporer. Data temporer adalah address dari VLR di mana pengguna tercatat, informasi *call forwarding*, dan parameter transien untuk autentikasi dan enkripsi.

2.3.9 *Visitor Location Register (VLR)*

VLR berisi data yang relevan dari semua stasiun bergerak yang sedang tercatat pada suatu area layanan. Data permanen ada di HLR. Data temporer agak berbeda, misalnya data dapat berisi TMSI. Bahkan sekalipun stasiun bergerak berada pada areanya sendiri, ia akan tercatat di VLR dan tentu saja HLR.

2.3.10 *Authentication Center (AUC)*

Semua algoritma autentikasi dan parameter-parameternya disimpan di AuC. AuC menyediakan untuk HLR atau VLR parameter-parameter yang diperlukan untuk mengautentikasi identitas pengguna. AuC mengetahui algoritma yang mana dan paremeter yang harus dipakai untuk pengguna tertentu. *SIM card* yang diberikan kepada pengguna berisi algoritma dan parameter yang sama dengan yang ada pada AuC

2.3.11 Konfidensialitas Identitas Pengguna

Sebelum penggguna membuat panggilan atau mulai *standby* untuk menerima panggilan, identitasnya harus diketahui oleh jaringan.

IMSI (*International Mobile Subscriber Identity*) secara unik mengidentifikasikan pelanggan. Biasanya yang dikirim adalah identitas temporer TMSI (*Temporary Mobile Subscriber Identity*), bukan IMSI. Ini dilakukan untuk mencegah *intruder*:

a. Memperoleh informasi mengenai sumberdaya yang sedang digunakan pengguna .
b. Mencegah pelacakan lokasi pengguna .
c. Mempersulit pencocokan data pengguna dengan data yang dikirimkan.

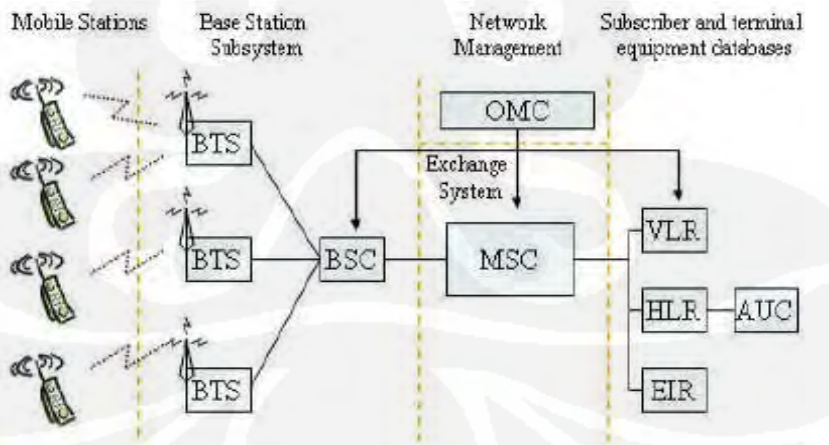
IMSI hanya dikirimkan bila diperlukan, misalnya ketika pengguna mengggunakan *SIM card* nya untuk saat pertama kali atau ada kehilangan data di VLR. Ketika SIM

card digunakan pertama kali, MS(*Mobile Station*) membaca *TMSI default* yang disimpan pada *card*. Kemudian MS mengirim TMSI *default* ini ke VLR. Karena VLR tidak tahu adanya TMSI ini, ia akan meminta IMSI dari MS. MS mengirim IMSI ke VLR. Kemudian VLR memberikan TMSI baru bagi pengguna tersebut. VLR mengirim TMSI baru ke MS dalam bentuk terenkripsi. Algoritma enkripsinya adalah A5. Kunci enkripsi adalah Kc. MS mendekripsikan pesan dan memperoleh TMSI. Selanjutnya MS hanya menggunakan TMSI untuk mengidentifikasi dirinya. TMSI hanya berukuran 5 digit, dan unik dalam area lokasi MS bergerak.

LAI (*Location Area Identification*) dan TMSI secara unik mengidentifikasikan penggguna. VLR menyimpan LAI dan TMSI untuk tiap pengguna pada areanya. Sebuah TMSI baru akan dialokasikan untuk tiap prosedur *update* lokasi. Jika system tidak gagal beroperasi (i.e. beroperasi dengan baik), IMSI tidak dipakai lagi. VLR baru selalu memperoleh IMSI dari VLR lama dengan menggunakan TMSI lama dan LAI yang dikirim oleh MS.

2.4 Pengamanan Untuk GSM

Arsitektur GSM dapat dilihat pada ilustrasi Gambar 2.8 di bawah ini .



Gambar 2.8 Arsitektur GSM [6]

Dari Gambar 2.8 di atas, dapat dilihat bahwa dalam berkomunikasi, telepon seluler, *mobile station* (MS), memanfaatkan layanan jaringan melalui *base station subsystem*

yang terdiri dari beberapa *base transceiver station* (BTS) dan sebuah *base station controller* (BSC). BSC akan terhubung dalam manajemen jaringan operator GSM. Subsistem jaringan memanfaatkan basis data berikut untuk keperluan otentikasi dan keamanan :

a. *Home Location Register* (HLR), basis data yang menyimpan seluruh informasi administratif dari tiap pelanggan jaringan GSM yang terdaftar, lengkap dengan lokasi terkini (*current location*) dari MS.
b. *Visitor Location Register* (VLR), melacak MS yang berada di luar *home network*, sehingga jaringan dapat dengan mudah mendeteksi keberadaan MS tersebut.
c. *Equipment Identity Register* (EIR), berisi daftar *International Mobile Equipment Identity* (IMEI) yang dibolehkan untuk menggunakan layanan jaringan.
d. *Authentication Center* (AuC), basis data yang berisi: *International Mobile Subscriber Identity* (IMSI), *Temporary Mobile Subscriber Identity* (TMSI), *Location Area Identity* (LAI), *dan Authentication Key* (Ki).

Ada beberapa cara yang dipakai dalam upaya melakukan pengamanan komunikasi jaringan GSM, yaitu :

a. *Personal Identification Number* (PIN) pada MS.
b. Otentikasi pengguna layanan.
c. Enkripsi pada GSM.
d. Penggunaan TMSI

2.5 Kriptografi

Kriptografi dalam sejarahnya tercatat dipergunakan secara terbatas oleh bangsa Mesir 4000 tahun lalu. Kriptografi (*Cryptography*) berasal dari dua kata yaitu "*Crypto & graphy*" yang dalam sudut bahasa "*Crypto*" dapat diartikan rahasia (*secret*) dan "*graphy"* dapat diartikan tulisan (*writing*) jadi Kriptografi (*Cryptography*) dapat diartikan sebagai suatu ilmu atau seni untuk mengamankan pesan agar aman dan dilakukan oleh "*Cryptographer"*. Orang yang melakukan

enkripsi terhadap suatu pesan atau praktisi kriptogragi disebut *"Cryptographer"*.Sebuah pesan yang tidak disandikan atau dienkripsi disebut sebagai plaintext atau disebut juga sebagai cleartext. Sedangkan pesan yang telah disandikan dengan sebuah algoritma kriptografi disebut sebagai ciphertext.Proses untuk mengubah plaintext ke chipertext disebut encryption atau encipherment. Sedang proses mengubah chipertext ke plaintext disebut decryption atau decipherment.

Fasilitas untuk mengkonversikan sebuah plaintext ke ciphertext atau sebaliknya disebut *Cryptographic system* atau *Cryptosystem* dimana sistem tesebut terdiri dari algoritma–algorima tertentu yang tergantung pada sistem yang digunakan. Algoritma kriptografi (*cryptographic algorithm*) disebut *cipher* yang merupakan persamaan matematik yang digunakan dalam proses enkripsi dan deskripsi dimana proses tersebut diatur oleh satu atau lebih kunci kriptografi. Kunci-kunci tersebut secara umum digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung sistem yang digunakan.

Proses enkrisi dan deskrisi secara matematis diterangkan sebagai berikut :

$$EK\ (M) = C \text{ (Proses Enkripsi)} \quad (2.1)$$

$$DK\ (C) = M \text{ (Proses Deskripsi)} \quad (2.2)$$

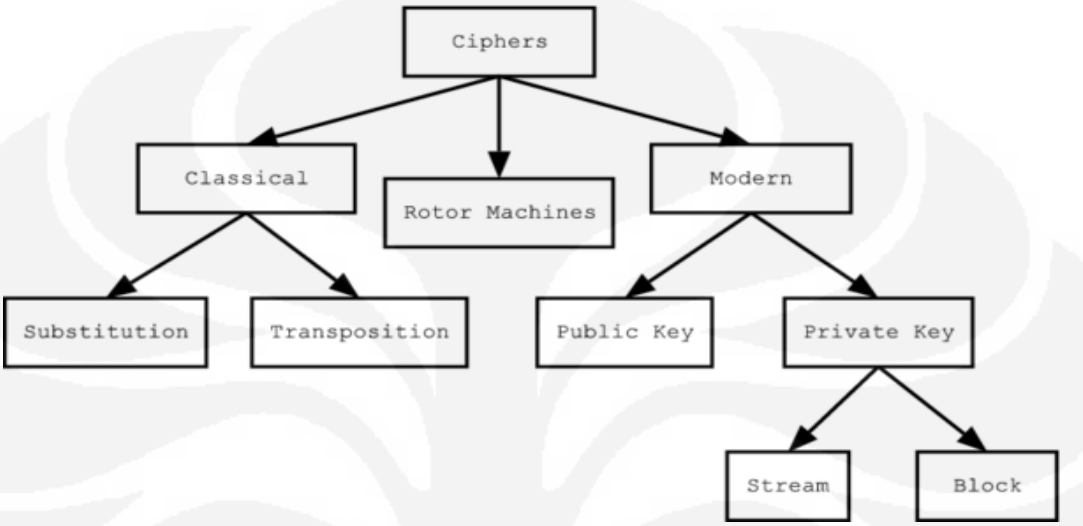
Keterangan :

EK : Enkripsi.
DK : Deskripsi.
M : Message (Pesan sebelum dienkripsi).
C : Cipher (Pesan setelah dienkrisi).

Secara umum algoritma kriptografi diciptakan oleh orang yang berpengalaman dalam bidang keamanan data dan mungkin pernah membuka sebuah algoritma kriptografi tanpa bantuan kunci. Pelaku yang melakukan tindakan memecahkan *ciphertext* tanpa

bantuan kunci disebut *Crytpanalyst*. Sedangkan Ilmu dan seni membuka (*breaking*) ciphertext tanpa bantuan kunci disebut *Cryptanalysis*.

Pada Gambar 2.9 di bawah ini, ditunjukkan ipe-tipe *Cipher* :



Gambar 2.9 Tipe *Cipher*

Tujuan dari adanya enkripsi adalah untuk meningkatkan keamanan data tetapi juga berfungsi untuk :

a. Melindungi data agar tidak dapat dibaca oleh orang-orang yang tidak berhak.
b. Mencegah agar orang-orang yang tidak berhak, menyisipkan atau mengahapus data.

Sedangkan tujuan dari sistem kriptografi adalah sebagai berikut :

a. *Confidentiality, m*emberikan kerahasiahan pesan dan menyimpan data dengan menyembunyikan informasi lewat teknik-teknik enkripsi.
b. *Message Integrity, m*emberikan jaminan untuk tiap bagian bahwa pesan tidak akan mengalami perubahan dari saat ia dibuat sampai saat ia dibuka.
c. *Non-repudiation, m*emberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang apabila ia mencoba menyangkal memiliki dokumen tersebut.

d. *Authentication, m*emberikan dua layanan. Pertama mengidentitifikasi keaslian suatu pesan dan memberikan jaminan keontentikannya. Kedua untuk menguji identitas seseorang apabila ia akan memasuki sebuah sistem.

Terdapat tiga kategori enkripsi yaitu :

a. Kunci enkripsi rahasia, dalam hal ini terdapat sebuah kunci yang digunakan untuk mengikrisi dan juga sekaligus mendeskripsikan informasi.
b. Kunci enkripsi *public*, dalam hal ini terdapat dua kunci yang digunakan, satu untuk proses enkripsi, satu lagi untuk proses deskripsi.
c. Fungsi *one-way*, dimana informasi dienkripsi untuk menciptakan "*signature*" dari informasi asli yang bisa digunakan untuk keperluan autentifikasi.

Dalam *Cryptosystem* menurut teknik enkripsinya dapat digolongkan menjadi dua buah, yaitu :

a. *Symmetric Cryptosystem* (Enkripsi Konvensional)

   Dalam *symmetric cryptosystem,*kunci yang digunakan dalam proses enkripsi dan dekripsi adalah sama atau pada prinsipnya identik. Kunci ini pun bisa diturunkan dari kunci lainnya. Oleh karena itu sistem ini sering disebut *secret-key ciphersystem.*

   Jumlah kunci yang dibutuhkan umumnya adalah :

$$ {}_nC_2 = \frac{n.(n-1)}{2} \qquad (2.1) $$

   Dimana n adalah banyaknya pengguna. Kunci yang menggunakan teknik enkripsi ini harus betul-betul dirahasiakan.

b. *Assymmetric Cryptosystem* (Enkripsi *public-key*)

   Dalam Assymmetric cryptosystem,kunci yang digunakan terdapat dua buah. Satu kunci yang dapat dipublikasikan deisebut kunci publik (*public key*), satu lagi kunci yang harus dirahasiakan disebut kunci privat (*private key*). Secara sedehana proses tersebut diterangkan sebagai berikut :

   a) A mengirimkan pesan kepada B.

b) A menyandikan pesannya dengan menggunakan kunci publik B.
c) Bila B ingin membaca pesan dari A, ia harus menggunakan kunci privatnya untuk mendekripsikan pesan yang tersandikan itu.

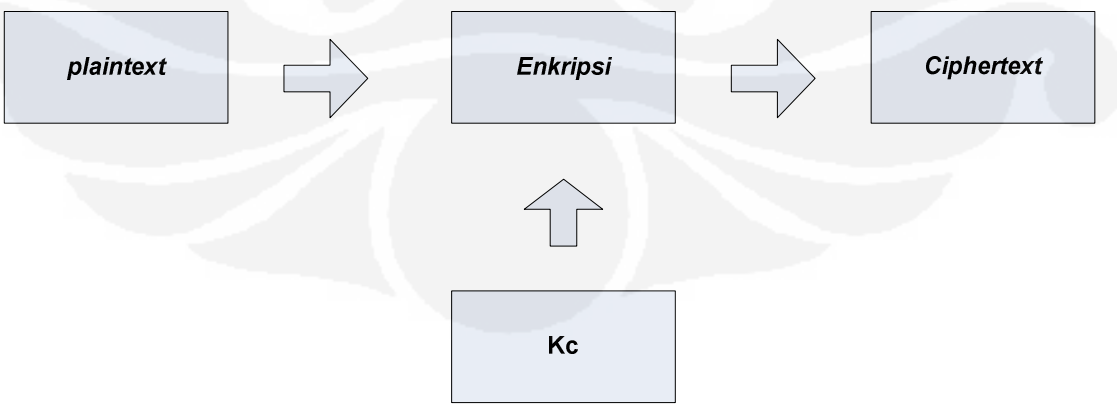
Dalam Cryptosystem yang baik harus memiliki karekteristik sebagai berikut :

a. Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang dipergunakan.
b. Cryptosystem yang baik memiliki ruang kunci (*keyspace*) yang besar.
c. Cryptosystem yang baik akan menghasilakn ciphertext yang terlihat acak dalam seluruh tes statistik yang dilakukan terhadapnya.
d. Cryptosystem yang baik mampu menahan seluruh serangan yang dikenal sebelumnya.

Bila salah satu hal diatas tidak dimiliki oleh sebuah *Cryptosystem* maka kemungkinan besar pesan yang di enkripsi oleh *Cryptosystem* dapat dibongkar sehingga isi dari pesan tersebut dapat dibaca oleh orang yang tidak berkepentingan. Salah satu *Cryptosystem* yang dapat dibongkar (*broken*) adalah LOKI 97, salah satu kandidat *Advanced Encryption Standard* (AES). LOKI 97 mempunyai kelemahan dalam persamaan matematika dan dalam f-functinnya.

2.5.1 Enkripsi

Secara singkat, pada Gambar 2.10 di bawah ini proses enkripsi adalah proses mengubah teks terang *plaintext* menjadi teks tersandi *ciphertext.*



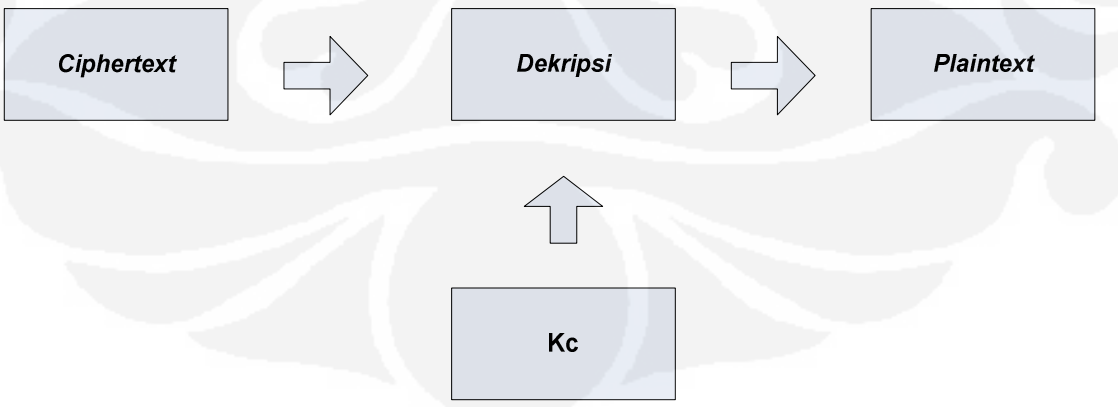
Gambar 2.10 Proses Enkripsi

Pada Gambar 2.10 di atas dijelaskan bahwa proses enkripsi dilakukan dengan penggabungan *plaintext* dengan Kc per byte untuk menghasilkan satu karakter. Di bidang kriptografi, enkripsi ialah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Dikarenakan enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Di pertengahan tahun 1970-an, enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah digunakan pada sistem secara luas, seperti Internet *e-commerce*, jaringan Telepon bergerak dan ATM pada bank.

Enkripsi dapat digunakan untuk tujuan keamanan, tetapi teknik lain masih diperlukan untuk membuat komunikasi yang aman, terutama untuk memastikan integritas dan autentikasi dari sebuah pesan. Contohnya, *Message Authentication Code* (MAC) atau *digital signature*. Penggunaan yang lain yaitu untuk melindungi dari analisis jaringan komputer.

2.5.2 Dekripsi

Pada Gambar 2.11 di bawah ini dilakukan proses dekripsi yang merupkan kebailikan dari proses enkripsi.



Gambar 2.11 Block Dekripsi

Pada proses dekripsi dilakukan dengan metode kebalikan dari proses enkripsi untuk diperolehnya nilai atau informasi awal. *Ciperhtext* dan Kc merupakan input untuk block dekripsi, di mana dilakukan proses komputasi sehingga akan diperoleh *pleintext.* Pada proses dekripsi ini, semua data dilakukan perhitungan invers dari perhitungan enkripsi, dan *ciphertext* yang merupakan informasi tersandi dapat dengan mudah dibaca atau diketahui informasinya setelah dilakukan proses dekripsi ini.

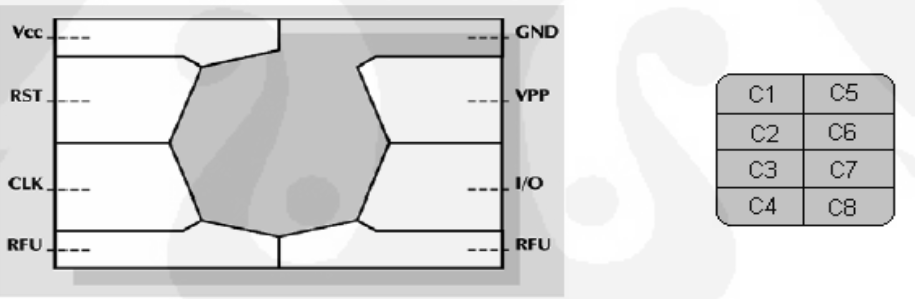
2.6 Konfigurasi Pin *Simcard.*

Memori pada kartu chip memiliki 2 fungsi dasar:

a. Tempat penyimpanan data.
b. Tempat menjalankan algoritma-algoritma untuk pembuktian identitas pelanggan.

Informasi data tersebut disimpan dalam file data untuk aplikasi khusus.

.



Gambar 2.12 Struktur kontak pada kartu [11]

Fungsi kontak-kontak pada Gambar 2.12 diatas adalah :

a. C1 digunakan untuk *input power supply* (Vcc) dari piranti antarmuka.
b. C2 untuk RST dan digunakan oleh piranti antarmuka untuk mengirim sinyal reset ke mikrosirkit kartu.
c. C3 untuk *clock* (CLK) dan sinyal-sinyal pewaktuan dikirimkan ke kartu melalui C3.
d. C5 sebagai tegangan referensi (GND), nilai tegangan itu dianggap 0 volt.
e. C6 secara bebas digunakan untuk memprogram atau menghapus (Vpp).

f. C7 menyelenggaraakan komunikasi ke dan dari kartu, dan disebut I/O.

g. C4 dan C8 tidak digunakan.

BAB 3
RANCANG BANGUN SISTEM

3.1 RANCANG BANGUN SOFTWARE

3.1.1 Diagram Alir

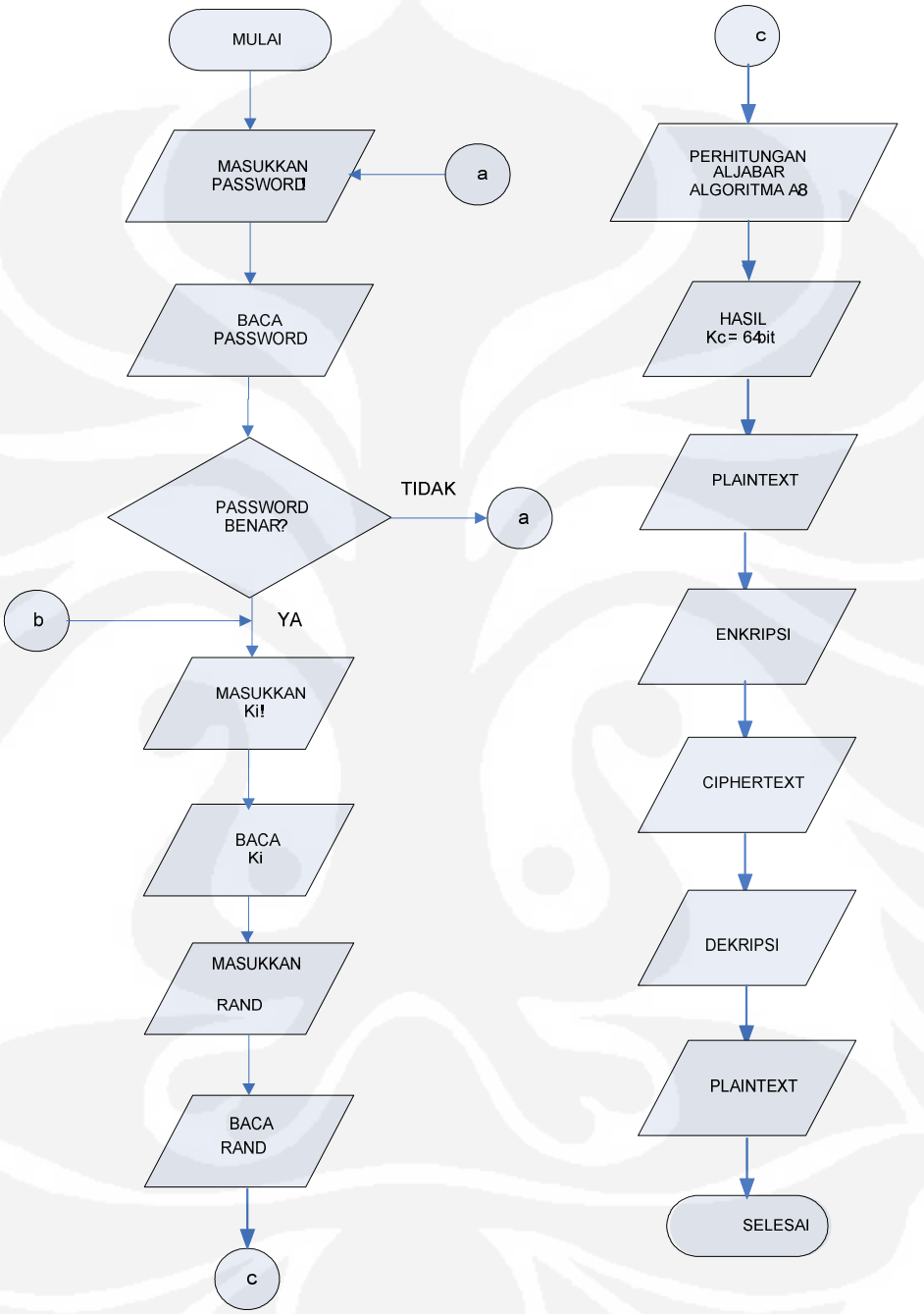
Pada Gambar 3.1 di bawah ini menunjukkan diagram alir dari proses pembangkitan kunci Kc awal hingga proses enkripsi dan dekripsi informasi. Dimana sistem menggunakan password sebagai PIN awal digunakannya simulasi, serta membutuhkan masukan berupa data hexadesimal sebesar 32 byte yang terdiri dari nilai data Ki dan RAND. Dan pada diagram alir tersebut, awal pertama simulasi memerlukan password yang benar sesuai dengan yang telah diprogramkan oleh user.

Kemudian dilanjutkan dengan permintaan untuk memasukkan nilai data Ki sebesar 32 *byte hexadecimal* dilanjutkan dengan memasukkan nilai data RAND sebesar 32 *byte hexadecimal* . Format tersebut diubah ke dalam byte untuk dapat dilakukan perhitungan dalam algoritma. Perancangan software tidak hanya melingkupi pada sisi pembangkitan kunci *ciphering key*, juga dilakukan proses tahap selanjutnya, yaitu dengan enkripsi data informasi yang dikirim, diubah menjadi acak dengan memanfaatkan kunci Kc *ciphering key* hingga dihasilkannya *ciphertext*.

Kemudian dalam proses *ciphering* ini, maka data informasi dikirim ke MS melalui MSC, di mana data informasi tersebut tidak langsung diterima oleh MS tujuan, karena data informasi tersebut masih dalam kondisi acak oleh algoritma A5 dengan masukan kunci *ciphering key* tersebut di atas.

Setelah di acak, maka data informasi tersebut di atas akan dikembalikan ke kondisi semula yaitu dalam kondisi di mana data informasi tersebut belum dikirim. Gambar 3.1 menunjukkan keseluruhan proses awal hingga akhir sistem.

Pada Gambar 3.1 Diagram alir di bawah ini, *plaintext* yang dikirim merupakan data awal yang nantinya akan dilakukan perhitungan dengan kunci *ciphering key*.



Gambar 3.1 Diagram Alir Proses Kc dan Pengamanan Komunikasi

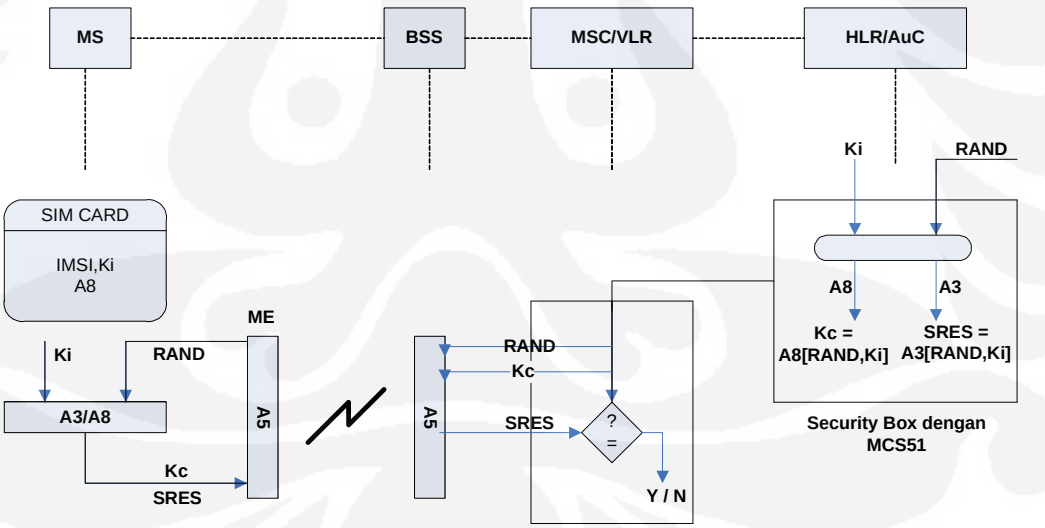
Pada Gambar 3.1 tersebut di atas, alur kerja sistem pembangkitan *chipering key* Kc harus menghasilkan kapasitas kunci sebesar 64 bit. Sedangkan untuk dapat

menghasilkan kunci Kc *ciphering key* dengan memasukkan data Ki dan RAND masing-masing 128 bit ke dalam simulasi program.

Data 128 bit dapat diubah dari hexadesimal ke dalam bentuk byte. Kemudian akan dihitung apakah nilai Ki dan RAND itu sama dengan 128 bit *fixed*. Bila tidak maka program akan *error* hingga besarnya kapasitas Ki dan RAND mencapai 128 bit. Proses pembangkitan kunci *ciphering key* menggunakan algoritma A8 sedangkan proses pengacakan data informasi menggunakan

3.1.2 Diagram Blok

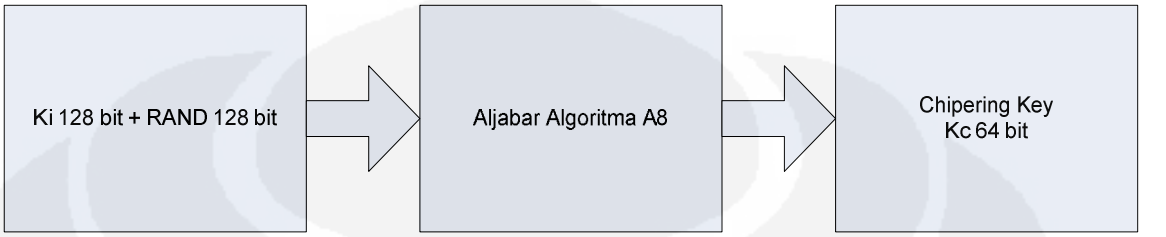
Software ini dirancang dengan menggunakan perhitungan algoritma A8. Di mana membutuhkan masukan Ki dan RAND sebagai proses awal dari komputasi ini. Pada rancang bangun software simulasi proses pembangkitan kunci *chipering key* Kc sebesar 64 bit ini membutuhkan masukan Ki 128 bit dan RAND 128 bit. Pada Gambar 3.2 di bawah ini, diperlihatkan bagian-bagian dari proses keamanan komunikasi GSM.



Gambar 3.2 Proses Keamanan Komunikasi GSM

Pada Gambar 3.3 tersebut di bawah ini, seluruh masukkan Ki dan RAND dalam proses simulasi akan diubah menjadi bilangan data desimal dari format data hexadesimal. Agar dapat digunakan parameter-parameter persamaan untuk

komputasi data atau perhitungan data dalam proses kompresi hingga lima *level* serta dilanjutkan dengan proses permutasi dari hasil akhir data yang telah dikompresi. Dengan demikian akan terbangkitnya kunci Kc.



Gambar 3.3 Blok Urutan Pembangkitan Kc

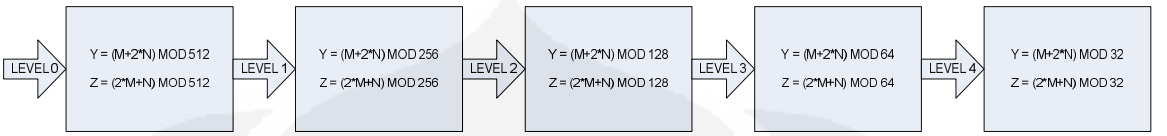
Pada Gambar 3.3 tersebut di atas, dijelaskan bahwa algoritma A8 dengan masukan Ki dan RAND dapat diproses pada blok aljabar A8, yang mana tahap ini adalah tahap awal untuk mendapatkan *chipering key* Kc. Data bit Ki dan RAND sebagai input diurutkan untuk bisa menjadi 256 bit data biner, seperti yang diperlihatkan pada Gambar 3.4 di bawah ini.

Gambar 3.4 *Blok Diagram* Algoritma Aljabar A8

Pada Gambar 3.4 tersebut di atas diperlihatkan bahwa Ki dengan 128 bit data beserta RAND yang juga sama sebesar 128 bit data dimasukan secara urut dalam bentuk data biner bit yang bila dijumlahkan menjadi 256 bit data biner. Kemudian data 256 bit ini dikompresi hingga 128 bit atau 32 byte. Dan data 128 bit ini dipermutasi menggunakan rumus hingga diperoleh *chipering key* Kc 64 bit.

Pada blok kompresi , data 256 bit yang sudah dimasukkan, akan melalui beberapa tahapan atau *level*. Tahapan yang pertama hingga ke lima menggunakan rumus Y = (M+2*N) MOD n yang dilanjutkan dengan urut Z = (2*M+N) MOD n. Rumus Y dan Z adalah pasangan rumus yang selalu digunakan pada setiap *looping* dari bit 0 hingga

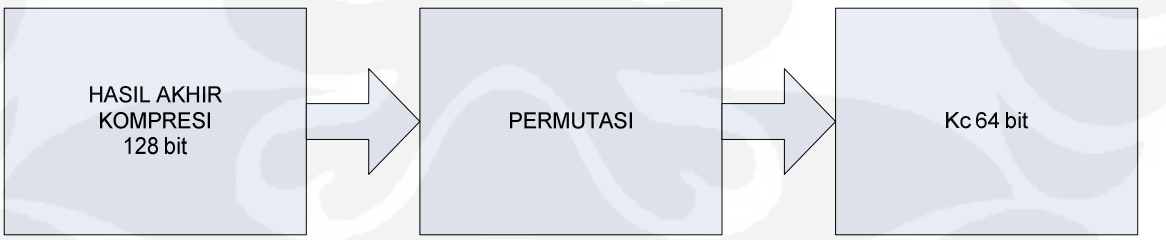
7 pada matrix X[0] hingga X[31]. Gambar 3.5 memperlihatkan tahapan level kompresi.



Gambar 3.5 Tahapan Level Kompresi A8

Pada Gambar 3.5 tersebut di atas, terdapat lima tahap kompresi data 256 bit Ki dan Rand, dengan tahapan level untuk modulus 512, 256, 128, 64 dan 32 diberikan kepada setiap rumus Y dan Z. Rumus Y = (M+2*N) modulus n (3.1)
dan rumus Z = (2*M+N) modulus n (3.2)
dimana untuk nilai n adalah berurutan untuk setiap *looping* 0 sampai 7 yaitu modulus 512, 256, 128, 64 dan 32. Dan hasil akhir setelah memasuki tahap *level* ke lima dengan modulus 32, maka total bit Ki dan RAND yang sudah diproses akan berjumlah 128 bit dari 256 bit yang masuk.

Setelah *level* ke lima selesai, maka data sebesar 128 bit ini akan diproses pada blok permutasi. Seperti pada Gambar 3.6 di bawah ini.



Gambar 3.6 Proses Akhir 128 bit Hasil Kompresi

Pada Gambar 3.6 tersebut di atas diketahui bahwa hasil akhir kompresi 128 bit akan dilakukan proses permutasi, di mana seluruh bit 128 diurutkan dari 0 hingga 127 yang akan dihitung dengan aljabar permutasi dengan memasukkan rumus :
j = [0....31]
i = [0.....7]

Yn = [8*j[x] + i]*17 modulus 128 (3.3)

Gambar 3.7 di bawah ini menunjukkan proses akhir dari pembangkitan *chipering key* Kc untuk tahapan setelah proses permutasi di atas. Untuk mendapatkan Kc, maka dipilihlah 4 bit lsb (*least Significan bit*) dari 1 *byte* .

X[0…31]

0	17	34	51	68	85	102	119

Gambar 3.7 Pengambilan bit Lsb

Pada Gambar 3.7 tersebut di atas, diambillah 4 bit dari sebelah kanan, pada urutan 68;85;102 dan 119. Kemudian urutan tersebut diperlukan untuk melakukan pengambilan bit dari 0....127 bit akhir kompresi, seperti Gambar 3.12 di bawah ini.

Pada Gambar 3.8 di bawah ini, diperluhatkan ketukan bit yang diambil berdasarkan pada persamaan (3.3) dalam tahap permutasi.

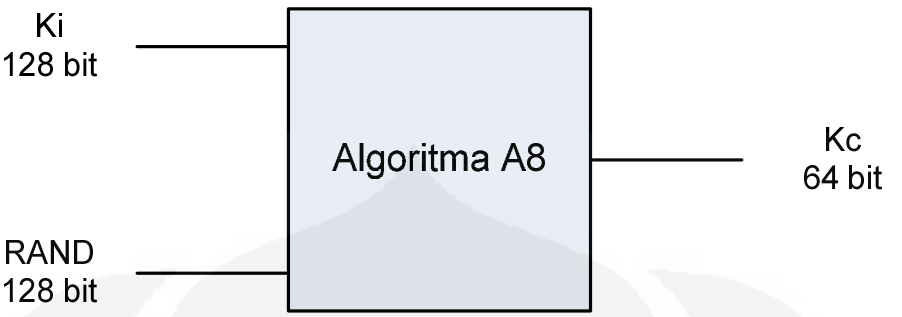
68	85	102	119
0	0	0	1

Gambar 3.8 Proses pengambilan data

Pada Gambar 3.8 tersebut di atas, proses pengambilan data bit 0 sampai 128 didasarkan pada bilangan yang sudah di dapat yaitu 68; 85; 102 dan 119. Dan hal ini terus dilakukan hingga X[31]. Dengan demikian akan didapatlah hasil akhir *chipering key* sebesar 64 bit atau 8 byte.

3.1.3 Algoritma

Dari rangkaian Gambar 3.9 di bawah ini, adalah rangkaian sistem pembangkitan kunci *ciphering key* menggunakan algoritma A8.



Gambar 3.9. Algoritma A8 Untuk Proses Pembangkitan Kunci Kc

Pada Gambar 3.9 tersebut di atas setiap level kompresi pada menghasilkan byte tergantung dari kedua masukan byte sebelumnya dengan rumusan :

$$X = (Z[m] + 2 * Z[n]) \bmod 2^{(9-j)} [13] \quad (3.4)$$

$$Y = (2 * Z[m] + Z[n]) \bmod 2^{(9-j)} [13] \quad (3.5)$$

$$Z[m] = T_j[X]$$

$$Z[n] = T_j[Y]$$

Dimana :

X = Elemen data baru sisi kiri

Y = Elemen data baru sisi kanan

Z [m] = Elemen dari Ki (*Individual Subscriber Authentication Key*) sebesar maksimal 128 bit

Z [n] = Elemen dari Rand (*Random Number*) sebesar maksimal 128 bit

j = Level kompresi data (Memiliki 5 level kompresi data)

- Kedua masukan byte digunakan untuk menentukan index dari *lookup table*, masukan *lookup table* akan digunakan untuk memperbaharui hasil byte
- *Lookup table* digunakan oleh level i sebagai table Ti dan berisi 2^{9-i} dari nilai 8-i bit, contohnya dapat diketahui pada table 3.1 di bawaj ini.

Pada Tabel 3.1 di bawah ini ditunjukkan mengenai tabel kompresi yang dapat dilakukan panggilan untuk setiap nilai yang akan disubstitusikan oleh masing-masing tabel, sesuai dengan level pada saat itu.

Table 3.1 Hasil data kompresi menggunakan *Butterfly structure*

Level	Nama Table	Jumlah Masukan	Nilai
1	Tab512	$2^9 = 512$	(9-i) Value = 8 bit
2	Tab256	$2^8 = 256$	(8-i) Value = 7 bit
3	Tab128	$2^7 = 128$	(7-i) Value = 6 bit
4	Tab64	$2^6 = 64$	(6-i) Value = 5 bit
5	Tab32	$2^5 = 32$	(5-i) Value = 4 bit

Setelah dilakukan proses pembangkitan kunci Kc, maka akan diteruskan kepada proses pengacakan informasi dengan algoritma A5 sehingga, data informasi tidak dapat disadap oleh intruder, seperti yang ditunjukkan pada Gambar 3.10 di bawah ini.

Gambar 3.10 Algoritma A5 Untuk Proses Acak Data

Dari Gambar 3.10 tersebut di atas, *plaintext* akan dikombinasikan dengan Kunci Kc dalam proses enkripsi dan dekripsi yang mana keseluruhannya diperoleh dalam perhitungan A5.

Persamaan yang digunakan dalam perhitungan algoritma A5 adalah sebagai berikut.

$A \oplus B = C$ .....(3.6)

$C \oplus B = A$ .....(3.7)

Dari persamaan (3.6) dan (3.7) di atas, *ciphering key* B yang sudah dihasilkan oleh algorima A8, dilakukan peritungan kembali terhadap *plaintext* A. Hasli perhitungan pertama merupakan c*iphertext* C. Kemudian data dikirim dalam bentuk acak atau data telah ter-*cipher* untuk tidak bisa disadap oleh para intruder. Kemudian pada saat data sampai kepada MS tujuan, maka informasi dalam bentuk *ciphertext* C akan

diubah kembali menjadi *plaintext* A bila dilakukan perhitungan kembali terhadap kunci *ciphering key*.

3.1.4 Prosedural Program

Dengan mengacu pada diagram alir dari Gambar 3.1, terdapat tiga tahap utama dalam proses pembangkitan Kc dan enkripsi, yaitu tahap inisialisasi awal, dimana user diminta untuk memasukkan PIN sebagai bentuk keabsahan pemilikan SIM Card, setelah itu menuju proses tahapan perhitungan Ki dan RAND oleh algoritma A8 yang akan menghasilkan kunci Kc *ciphering key*, dan terakhir adalah tahap enkripsi dekripsi dengan menggunakan algoritma A5.

Tahapan - tahapan tersebut adalah sebagai berikut :

3.1.4.1 PIN (*Personal Identity Number*)

a= password
b= baca password
jika a==b
    jalankan program berikut
jika a=/=b
    tampilkan “masukkan password yang benar”
selesai

3.1.4.2 Algoritma A8

c,input Ki
d,input RAND
tampilkan “ level1”
tampilkan y,[ab bc cd de ef fg gh hi ij jk kl lm mn no op pq]
tampilkan x, [aab bbc ccd dde eef ffg ggh hhi iij jjk kkl llm mmn nno oop ppq]
tampilkan ”lookup Tab512”
lyy1,Tab512(ab)
tampilkan lxx,[lxx, lxx2, lxx3, lxx4, lxx5, lxx6, lxx7, lxx8, lxx9, lxx10, lxx11,12, lxx13, lxx14, lxx15, lxx16]
tampilan ”level kedua”
level1=[lyy; lxx]
*dy1,lyy1+2*lxx1;,*
tampilkan ”lookup Tab256”
dlyy1,Tab256(dyy1+2);, dlyy2,Tab256(dyy2+2);, dlyy3,Tab256(dyy3+2);, dlyy4,Tab256(dyy4+2);, dlyy5,Tab256(dyy5+2);, dlyy6,Tab256(dyy6+2);, dlyy7,Tab256(dyy7+2);,dlyy8,Tab256(dyy8+2);, dlyy9,Tab256(dyy9+2);,

dlyy10,Tab256(dyy10+2);, dlyy11,Tab256(dyy11+2);, dlyy12, Tab256(dyy12+1);, dlyy13,Tab256(dyy13+2);, dlyy14,Tab256(dyy14+2);, dlyy15 , Tab256(dyy15+2);, dlyy16,Tab256(dyy16+2);
tampilkan "hasil level kedua'
dlyy , [dlyy1 dlyy2 dlyy3 dlyy4 dlyy5 dlyy6 dlyy7 dlyy8 dlxx1 dlxx2 dlxx3 dlxx4 dlxx5 dlxx6 dlxx7 dlxx8];
dlxx , [dlyy9 dlyy10 dlyy11 dlyy12 dlyy13 dlyy14 dlyy15 dlyy16 dlxx9 dlxx10 dlxx11 dlxx12 dlxx13 dlxx14 dlxx15 dlxx16];
tampilkan 'level ketiga'
lev3,[dlyy1 dlyy2 dlyy3 dlyy4; dlyy5 dlyy6 dlyy7 dlyy8; dlxx1 dlxx2 dlxx3 dlxx4; dlxx5 dlxx6 dlxx7 dlxx8;dlyy9 dlyy10 dlyy11 dlyy12; dlyy13 dlyy14 dlyy15 dlyy16;dlxx9 dlxx10 dlxx11 dlxx12; dlxx13 dlxx14 dlxx15 dlxx16]
tampilkan 'hasil level ke tiga'
ty,[tyy1 tyy2 tyy3 tyy4 txx1 txx2 txx3 txx4 tyy5 tyy6 tyy7 tyy8 txx5 txx6 txx7 txx8]
tx,[tyy9 tyy10 tyy11 tyy12 txx9 txx10 txx11 txx12 tyy13 tyy14 tyy15 tyy16 txx13 txx14 txx15 txx16]
tampilkan 'lookup Tab128'
tlyy1 , Tab128(tyy1+1);, tlyy2 , Tab128(tyy2+1);, tlyy3 , Tab128(tyy3+1);, tlyy4 , Tab128(tyy4+1);, tlyy5 , Tab128(tyy5+1);, tlyy6 , Tab128(tyy6+1);, tlyy7 , Tab128(tyy7+1);, tlyy8 ,
tampilkan 'Hasil Lookup Tab128'
tlyy , [tlyy1 tlyy2 tlyy3 tlyy4 tlxx1 tlxx2 tlxx3 tlxx4 tlyy5 tlyy6 tlyy7 tlyy8 tlxx5 tlxx6 tlxx7 tlxx8]
tlxx , [tlyy9 tlyy10 tlyy11 tlyy12 tlxx9 tlxx10 tlxx11 tlxx12 tlyy13 tlyy14 tlyy15 tlyy16 tlxx13 tlxx14 tlxx15 tlxx16]
tampilkan "level kelima"
tampilkan "hasil level keempat"
ey,[eyy1 eyy2 eyy3 eyy4 exx1 exx2 exx3 exx4 eyy5 eyy6 eyy7 eyy8 exx5 exx6 exx7 exx8]
ex,[eyy9 eyy10 eyy11 eyy12 exx9 exx10 exx11 exx12 eyy13 eyy14 eyy15 eyy16 exx13 exx14 exx15 exx16]
tampilkan "lookup table 64
elyy1 , Tab64(eyy1+1);, elyy2 , Tab64(eyy2+1);, elyy3 , Tab64(eyy3+1);, elyy4 , Tab64(eyy4+1);, elyy5 , Tab64(eyy5+1);, elyy6 , Tab64(eyy6+1);, elyy7 , Tab64(eyy7+1);, elyy8 , Tab64(eyy8+1);, elyy9 , Tab64(eyy9+1);, elyy10 ,
tampilkan "Hasil Lookup Tab64"
elyy , [elyy1 elyy2 elxx1 elxx2 elyy3 elyy4 elxx3 elxx4 elyy5 elyy6 elxx5 elxx6 elyy7 elyy8 elxx7 elxx8]
elxx , [elyy9 elyy10 elxx9 elxx10 elyy11 elyy12 elxx11 elxx12 elyy13 elyy14 elxx13 elxx14 elyy15 elyy16 elxx15 elxx16]
Tampilkan "level kelima"
lev5 , [elyy1; elyy2; elxx1; elxx2; elyy3; elyy4; elxx3; elxx4; elyy5; elyy6; elxx5; elxx6; elyy7; elyy8; elxx7; elxx8; elyy9; elyy10; elxx9; elxx10; elyy11; elyy12; elxx11; elxx12; elyy13; elyy14; elxx13; elxx14; elyy15; elyy16; elxx15; elxx16]
Tampilkan "hasil rumus level ke lima"
Tampilkan "lookup table 32"
Llyy1 , Tab32(Lyy1+1);
Tampilkan "Hasil Lookup Tab32"

Llyy , [Llyy1 Llxx1 Llyy2 Llxx2 Llyy3 Llxx3 Llyy4 Llxx4 Llyy5 Llxx5 Llyy6 Llxx6 Llyy7 Llxx7 Llyy8 Llxx8];
Tampilkan ”hasil kompresi akhir”
Kompress ,[Llyy1 Llxx1 Llyy2 Llxx2 Llyy3 Llxx3 Llyy4 Llxx4 Llyy5 Llxx5 Llyy6 Llxx6 Llyy7 Llxx7 Llyy8 Llxx8 Llyy9 Llxx9 Llyy10 Llxx10 Llyy11 Llxx11 Llyy12 Llxx12 Llyy13 Llxx13 Llyy14 Llxx14 Llyy15 Llxx15 Llyy16 Llxx16];
tampilkan “permutasi’
tampilkan ”ketukan untuk setiap bit level kompresi akhir”
tampilkan ”hasil pemrmutasi”

3.1.4.3 Algoritma A5

tampilkan “proses enkripsi dekripsi”
tampilkan ”nilai ciphering key”
Tampilkan “Kc”
Kc2,ubah Kc ke desimal
hc,mod(Kc2,127);
tampilkan “Informasi Yang Dikirim”
info, masukkan data
tampilkan ”Nilai Kc”
desimal,double(info);
enkrip,xor (decimal,Kc);
ciphertext,tampilkan plaintext enkripsi
dekripsi,xor (enkripsi,Kc);
plaintext ,tampilkan plaintext enkripsi

3.2 PERANCANGAN *HARDWARE*

Sistem perangkat keras *(hardware)* pada sistem ini dapat dibagi menjadi 3 bagian, yaitu bagian sistem pengolah informasi, bagian sistem simulasi, dan bagian sistem penampil *(display).* Masing-masing bagian sistem tersebut terdapat beberapa komponen pendukung dimana komponen pendukung tersebut mempunyai fungsi menurut bagiannya sendiri-sendiri.

3.2.1 Rangkaian Catu Daya

Power Supply yang dirancang untuk sistem kerja alat ini menggunakan trafo CT 1 Ampere, sebagai pengaman juga digunakan fuse pada sisi trafo, sehingga apabila ada masalah pada sistem maka akan otomatis memutuskan supply untuk mencegah kerusakan pada sisi lilitan maupun rangkaian penyearah.

Tegangan output regulator 12VDC dan 9 VAC dihasilkan oleh catu daya pada Gambar 3.11 di atas. Untuk tegangan 12 VDC digunakan untuk memberikan supply pada kipas sebagai pendingin, sedangkan tegangan 9 VAC digunakan untuk supply pada minimum sistem DT-51, karena pada modul DT-51 sudah memiliki sistem catu yang mengubah 9 VAC menjadi regulator 5 volt dc sehingga tegangan pada modul tersebut menjadi stabil walau tegangan inputnya naik turun.

3.2.2 Minimum Sistem DT-51

Mikrokontroller yang digunakan pada modul minimum sistem DT-51 Ver. 3. 3, menggunakan mikrokontroller tipe AT89S52. Memori eksternal RAM dengan kapasitas memory 64 Kbyte (28HC64) dan PPI 8255 (*Programmable Peripheral Interface*). Pada PPI 8255 ini terdapat 4 *Port* sebagai *interface* data bus. Ke-empat *port* tersebut adalah:

a. *Port* A, port ini digunakan sebagai *output* (*address* 2000H)
b. *Port* B, port ini digunakan sebagai *input* (*address* 2001H)
c. *Port* C dan *Port* 1 digunakan sebagai *output* (*address* 2002H)
d. *Port Control Word Register* (2003H)

Dengan penjelasan sebagai berikut :

a. *Port* A, *port* ini digunakan sebagai *output* (*address* 2000H)

Out &H2000 , A1

A1 merupakan *register* yang digunakan untuk mengeluarkan aplikasi pada *keypad*. Penggunaan *address* 2000H untuk mengaktifkan *port* A sebagai *output* pada *keypad* matriks 4x4, sehingga *keypad* tersebut dapat berfungsi sebagai *output* kolom data untuk masukan data simulasi.

b. *Port* B, *port* ini digunakan sebagai *input* (*address* 2001H)

B1 = Inp(&H2001)

B1 merupakan *register* yang digunakan untuk memberikan masukan aplikasi pada *keypad*. Penggunaan *address* 2001H adalah untuk mengecek bit data pada *port* B

sebagai *input* tombol dari *keypad* matriks 4x4, sehingga *keypad* tersebut dapat mampu membaca baris data pada *keypad*.

c. *Port* C dan *Port* 1 digunakan sebagai *output* (*address* 2002H)

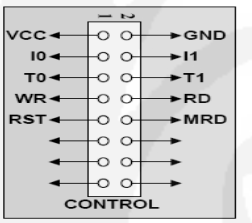
Penggunaan *port* C ini untuk mengeluarkan semua simulasi sensor dari *keypad* matriks 4x4 jika ada penambahan *output* simulasi berupa buzzer atau led.

d. *Port Control Word Register* (2003H)

Fungsi *port* ini untuk mengaktifkan *keypad* pada posisi Write, sehingga *keypad* bisa difungsikan sebagai *input.*

Misal : **Out &H2003 , &B10000010**

Ini berarti 8 bit yang difungsikan yaitu 10000010 bit referensi untuk mengaktifkan fungsi *port* pada kondisi ’write’. Bit ke -7 merupakan bit control untuk kondisi ‘write’. Dimana diagram *port control* dapat dilihat pada Gambar 3.2.



Gambar 3.12 Diagram *port control* [9]

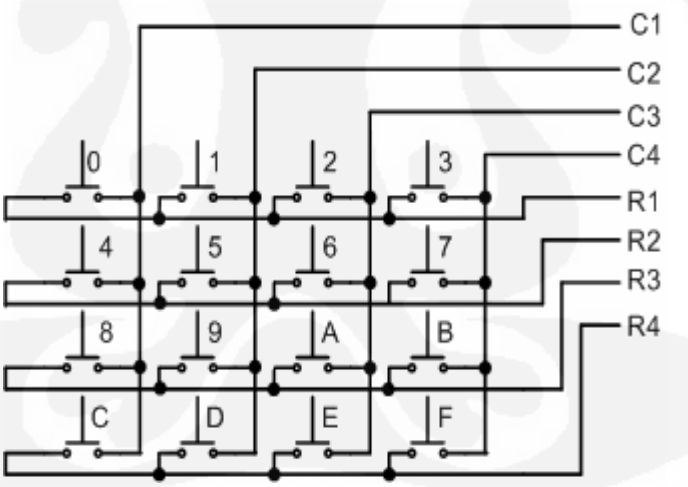
Di samping terdapat PPI 8255 (*programmable Peripheral Interface*), modul DT-51 juga memiliki RAM eksternal 28HC64 untuk penyimpanan data, selain bisa disimpan pada memory internal (0000H – 1FFFH) juga dapat disimpan pada memory eksternal (6000H – FFFFH). Dengan adanya memory eksternal ini maka dapat memudahkan penyimpanan data sementara *(temporary data storage)* pada pemrograman BASCOM IDE 8051. Penyimpanan data sementara dapat berupa data bit, byte, word, dan integer. Ukuran dari data tersebut adalah, bit memiliki ukuran data 0 dan 1, byte memiliki ukuran data 0 – 255, word memiliki ukuran data 0- 2047, integer memiliki ukuran data -32767 - +32768.

3.2.3 Perangkat *Input* Simulasi

Perangkat input simulasi terdiri dari beberapa perangkat pendukung, yang bertujuan memberikan masukan berupa data Ki dan RAND ke program kopmutasi data. Adapun perangkat pendukung digunakanlah keypad.

3.2.3.1 *Keypad*

Keypad yang digunakan adalah keypad matriks 4x4. Keypad pada sistem ini memiliki fungsi sebagai input data password dan input data untuk simulasi sensor. Perancangan program keypad untuk simulasi ini adalah dengan metode ’grounding’, maksudnya adalah bit akan dalam kondisi ’0’ ketika tombol keypad tersebut ditekan. Hal ini dilakukan karena pin-pin dari port PPI sudah dalam kondisi high atau kondisi ’1’ sebelumnya sehingga nilai dari semua port pada PPI tersebut adalah 255 (FFH = 11111111). Sehiingga dengan menggunakan metode grounding akan memudahkan pengecekan bit-bitnya.



Gambar 3.13 Hubungan keypad 4X4

3.2.3.2 Perangkat *Output* Simulasi

Fungsi dari perangkat output simulasi adalah menampilkan setiap hasil dari perubahan kondisi yang terjadi baik saat kondisi simulasi belum berjalan (masih

dalam kondisi password belum dibuka) maupun kondisi simulasi telah dijalankan (SRes mampu ditampilkan). Perangkat penampil informasi yang lazim digunakan adalah LCD (*Liquid Crystal Display*)

3.2.3.2.1 LCD (*Liquid Crystal Display*)

Pada perancangan untuk alat ini, LCD (*Liquid Crystal Display*) yang digunakan adalah LCD dengan ukuran 2x16 karakter. Semua pin-pin dari LCD (*Liquid Crystal Display*) tersebut harus terkoneksi dengan tepat pada modul DT-51. Kemudahan dari Modul ini adalah telah disediakannya port khusus untuk semua pin dari LCD, sehingga hal ini mampu mengurangi kesalahan dalam pemasangan pin-pin dari LCD. Pada sistem ini LCD (*Liquid Crystal Display*) difungsikan sebagai *output* untuk menampilkan semua instruksi-instruksi dan informasi yang berkaitan dengan sistem kerja alat ini, sehingga dengan adanya LCD ini tidak terjadinya kesalahan prosedur dalam pengaplikasiannya.

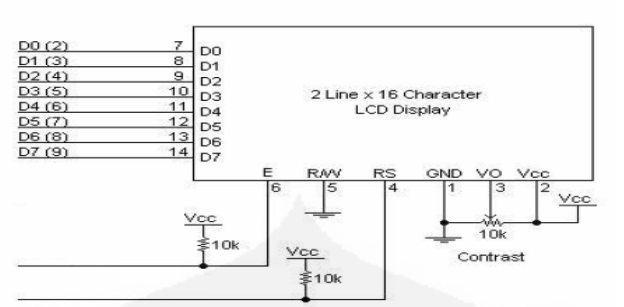
Hal-hal yang berkaitan dengan fungsi kerja dari LCD ini antara lain:

a. Menampilkan semua input dari keypad baik berupa data password maupun berupa data input simulasi.
b. Menampilkan informasi berupa status proses pencarian nilai triplet-triplet autentikasi.
c. Menampilkan proses komputasi dan output proses komputasi.

Adapun fitur yang disajikan dalam LCD ini adalah:

a. Terdiri dari 16 karakter dan 2 baris
b. Mempunyai 192 karakter tersimpan
c. Terdapat karakter generator terprogram
d. Dapat dialamati dengan mode 8-bit dan 4-bit
e. Dilengkapi dengan *backlight*

Pada Gambar 3.14 di bawah ini, LCD ini terdiri dari 8 jalur data, 3 jalur kendali dan fasilitas pengaturan kontras serta *backlight*. LCD ini dapat dikendalikan dengan mikrokontroller atau mikroposesor.



Gambar 3.14 LCD 16x2 [13]

Deskripsi fungsi-fungsi pin dari penampil LCD (*Liquid Crystal Display*) ini adalah:

a. VSS, sebagai suplai 0 volt atau ground.
b. VDD, sebagai suplai input untuk LCD (5 volt).
c. VO, sebagai tegangan operasi untuk LCD atau pengaturan kontras layer.
d. RS, sebagai jalur data dan penghantar kode instruksi. (H: DATA, L: kode instruksi).
e. R/W, sebagai jalur baca-tulis data ke mikroprosesor.
f. E, sebagai sinyal *enable* bagi *chip*.
g. DB0-DB7, sebagai data bit 0 sampai dengan data bit 7.

BAB 4

UJI COBA DAN ANALISIS

4.1 PROSEDUR SIMULASI

Uji coba simulasi ini dilakukan dengan tujuan untuk menganalisis karakteristik unjuk kerja pada rancangan algoritma A8 dalam proses pembangkitan kunci Kc dan enkripsi maupun dekripsi.

Uji coba simulasi ini terdiri dari dua bagian uji karakteristik unjuk kerja yaitu

- Uji karakteristik unjuk kerja algoritma A8.
- Uji karakteristik unjuk kerja proses pembangkitan Kc terhadap usaha intruder dalam menyadap informasi.

Struktur perangkat model simulasi terdiri dari mokrokontroller AT9S52, yang berfungsi sebagai pensimulasi penghitungan proses *chipering key generate*.

Tahap awal yang harus dilakukan adalah menentukan nilai Ki (yang di simulasikan dengan masukan dari keypad) sebesar maksimal 16 byte apabila kurang dari 16 byte algoritma tidak bisa melakukan proses simulasi komputasi ke langkah berikutnya. Sesaat setelah akses dilakukan, proses autentikasi segera dilaksanakan.

Pengolahan Algoritma A8 dijalankan dengan beberapa tahapan;

1. Tahap pertama, melakukan kompresi data Ki dan RAND hingga lima level kompresi dengan demikian maka nilai level terakhir menghasilkan nilai bit yang semakin sederhana.
2. Tahap kedua, algoritma A8 menjalankan proses pembentukan matrik untuk permutasi dari hasil perubahan byte menjadi bit dari hasil kompresi nilai data ke lima level menjadi nila bit.
3. Tahap ketiga, bit-bit yang dihasilkan dari nilai data kelima level akan membentuk matrik-matrik baru untuk dilakukan proses permutasi dengan mode yang telah dihitung, sehingga akan menghasilkan Kc sebesar 64 bit.

4.2 HASIL UJI COBA SIMULASI DAN ANALISIS

4.2.1 Karakteristik Unjuk Kerja Algoritma A8 dan Analisis

Uji coba karakteristik unjuk kerja dari rancangan algoritma A8 dilakukan untuk mengetahui jumlah kapasitas keluaran yang digunakan dari kedua masukan.
Untuk itu, maka uji coba dilakukan dengan menggunakan beberapa Ki yang berbeda nilainya satu sama lain.

4.2.1.1. Unjuk kerja algoritma A8 berdasarkan perhitungan menggunakan PC berbasis program tertentu.

DATA PERCOBAAN 1

Pada Tabel 4.1 di bawah ini, diakukan proses pengubahan atau konversi format hexadecimal.

Tabel 4.1 Konversi Desimal ke Hexadesimal

No.	Ki	RAND
1	20 20 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	C C 17 2D 41 4C 57 62 2B 2B 36 41 4C 57 62 20
2	22 23 23 20 17 41 21 2D 43 C 22 37 43 58 62 21	22 2B 36 38 43 57 62 15 20 22 36 41 56 36 57 62
3	C 17 22 2D 38 43 4E 59 5A 62 57 4C 41 36 2B 20	1F 20 2B 36 20 36 41 4C 20 2D 41 17 2D 42 4E 63
4	22 20 38 43 22 2D 38 17 22 2D 38 4E 59 57 57 41	C 17 2B 2D 41 4C 57 62 C D E E 36 20 41 4C
5	20 22 15 40 20 4B 56 20 37 15 41 57 61 61 17 C	20 22 15 40 20 4B 56 20 37 15 41 57 61 61 17 C
6	22 36 2B 4C 58 5A 15 20 62 41 2B 62 57 2B 20 2B	22 2D 38 43 22 2D 38 17 22 2D 38 4E 59 57 57 41
7	C 17 2B 2D 41 4C 57 62 C D E E 36 20 41 4C	22 2D 38 43 22 2D 38 17 22 2D 38 4E 59 57 57 41
8	1F 20 2B 36 20 36 41 4C 20 20 41 17 2D 42 4E 63	22 23 23 2D 17 41 21 2D 43 C 22 37 43 58 62 21
9	22 2B 36 28 43 57 62 15 20 22 36 41 56 36 57 62	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
10	C C 17 2D 41 4C 57 62 2B 2B 36 41 4C 57 62 20	22 36 2B 4C 58 5A 15 20 62 41 2B 62 57 2B 20 2B

Pada Tabel 4.2 di bawah, diperlihatkan bahwa seluruh input Ki maupun RAND diubah ke dalam bentuk desimal untuk bisa dihitung dengan mudah menggunakan algoritma A8 berbasis PC.

Tabel 4.2 di bawah ini diperlihatkan hasil komputasi dari bermacam-macam Ki dan RAND yang berbeda diubah dalam bentuk desimal.

Tabel 4.2 Uji Coba Ki dan RAND Berubah.

No	Ki	RAND
1	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	12 12 23 45 65 76 87 98 43 43 54 65 76 87 98 32
2	34 35 35 45 23 65 33 45 67 12 34 55 67 88 98 33	34 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98
3	12 23 34 45 56 67 78 89 90 98 87 76 65 54 43 32	31 32 43 54 32 54 65 76 32 45 65 23 45 66 78 99
4	34 45 56 67 34 45 56 23 34 45 56 78 89 87 87 65	12 23 43 45 65 76 87 98 12 13 14 14 54 32 65 76
5	32 34 21 64 32 75 86 32 55 21 65 87 97 97 23 12	32 34 21 64 32 75 86 32 55 21 65 87 97 97 23 12
6	34 54 43 76 88 90 21 32 98 65 43 98 87 43 32 43	34 45 56 67 34 45 56 23 34 45 56 78 89 87 87 65
7	12 23 43 45 65 76 87 98 12 13 14 14 54 32 65 76	34 45 56 67 34 45 56 23 34 45 56 78 89 87 87 65
8	31 32 43 54 32 54 65 76 32 45 65 23 45 66 78 99	34 35 35 45 23 65 33 45 67 12 34 55 67 88 98 33
9	34 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
10	12 12 23 45 65 76 87 98 43 43 54 65 76 87 98 32	34 54 43 76 88 90 21 32 98 65 43 98 87 43 32 43

Dan pada Tabel 4.2 tersebut di atas, setelah diubahnya format seluruh Ki dan RAND menjadi desimal, maka dilakukanlah komputasi perhitungan Kc oleh algoritma A8, di mana hasilnya diperlihatkan pada Tabel 4.3 di bawah ini. Pada Tabel 4.3 di bawah ini, diperlihatkan hasil keseluruhan percobaan Ki dan RAND yang dinputkan dengan nilai yang selalu dinamis, dan terbukti bahwa perhitungan Ki dan RAND akan menghasilkan jumlah bit Kc sebesar 64 bit. Setiap Ki dan RAND yang berbeda akan selalu menghasilkan Kc yang berbeda.

Dan tahapan berikutnya setelah dilakukan inputan Ki dan RAND maka akan dihasilkan Kc seperti pada Tabel 4.3 di bawah ini.

Tabel 4.3 Hasil Kc (RAND dan Ki Berubah)

No	Kc
1	1110001001101111000111110011000011110010000011010110000000000000
2	1001110100111111101100100000101111100101011001000000110000000000
3	1001001011100010010100100111111100011011101010000011100000000000
4	0101101101000111000110111001001101010010011110010010110000000000
5	0110100010000100000011110011111000101100110001000000010000000000
6	1011111011101010000010101111100100010101110101011100000000000000
7	0001011011110000101001100000100000011111100011100001100000000000
8	0101111010011000010100010001100011001101000000000100100000000000
9	0100110111111110101111011011011101011110011100110010010000000000
10	1100000010100010111011001010001101001111011110101111100000000000

Perhitungan Kc ini menggunakan aturan algoritma A8, yang mana digunakan aljabar sebagai berikut :

lev1 = input ('masukkan nilai Ki;RAND @16 byte = ')

Syntax di atas digunakan untuk melakukan penerimaan input Ki dan RAND ke dalam system algoritma A8.

yn=lev1(m,k)+2*lev1(m,k); (4.1)

yx=lev1(m,k)*2+lev1(m,k); (4.2)

di mana :

n = data ke satu hingga ke-16

m = baris urutan bit array ke m

k = kolom urutan bit array ke k

Pada persamaan 4.1 di atas diterangkan bahwa difungsikan untuk melakukan perhitungan awal untuk level ke satu kompresi, di mana input awalnya merupakan Ki dan RAND.

yyn= mod(yn,j); (4.3)

xxn= mod(yn,j); (4.4)

di mana :

n = data ke satu hingga ke-16

j = modulus 512, 256, 128, 64 hingga 32.

Pada persamaan 4.3 dan 4.4 di atas, dijelaskan bahwa data yang telah dihitung oleh persamaan 4.1 dan 4.2 dilanjutakan dengan melakukan modulus terhadap variabel *yn* dan variabel *xn* sesuai dengan level yang sedang dilalui. Level kompresi ini terdapat lima tahapan dengan modulus yang berbeda, yaitu modulus 512, modulus 256, modulus 128, modulus 64 dan modulus 32.

y=[yy1 ... yy15] (4.5)

x=[xx1 ... xx16] (4.6)

Pada persamaan 4.5 dan 4.6 dilakukan pembentukkan array baru dari hasil perhitungan persamaan 4.1, 4.2, 4.3 dan 4.4 di atas.

Kemudian dilakukanlah proses substitusi tabel GSM sesuai dengan level kompresi yang dijalani pada saat itu. Karena ada lima tahap kompresi, maka jumlah tabel GSM juga ada lima bagian, yaitu Tab512, Tab256, Tab128, Tab64 dan Tab32, seperti yang ditunjukkan pada persamaan 4.7 di bawah ini.

lyyn = Tabj(yyn) (4.7)

lyyx = Tabj(yyx) (4.8)

di mana :

n = 1...16

x = 1...16

j = 512, 256, 128, 64, 32.

Dan hasil persamaan 4.7 dan 4.8 di atas digunakan untuk akhir kompresi pada setiap tahap hingga tahap terakhir dari level kompresi, di mana nilai yang dikompres menggunakan substitusi tabel GSM, sehingga berubah menjadi jumlah bit yang semakin kecil nilainya.

Begitu juga proses di atas dilakukan sama untuk menghasilkan Kc pada Tabel 4.3 ,

DATA PERCOBAAN 2

Pada Tabel 4.4 di bawah ini, digunakanlah uji coba dengan menggunakan masukan Ki dan RAND yang berbeda dari percobaan sebelumnya. Konversi dari desimal ke hexadesimal diperlihatkan pada Tabel 4.4 di bawah. Pengkonversian ini menunjukkan bahwa program untuk mikrokontroller membutuhkan masukkan dalam format hexadesimal. Sedangkan untuk masukkan ke dalam program yang berbasis pada PC, menggunakan format byte, yang kesemuanya adalah tergantung dari kemudahan syntax yang terdapat pada setiap program dan *device*.

Tabel 4.4 Konversi Desimal ke Hexadesimal

No.	Ki	RAND
1	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	C C 17 2D 41 4C 57 62 2B 2B 36 41 4C 57 62 20
2	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	2B 2B 36 38 43 57 62 43 2B 36 41 4C 57 62 20
3	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	1F 20 36 38 43 57 62 15 20 22 36 41 56 36 57 62
4	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	C 17 2B 2D 41 4C 57 62 C D E E 3B 20 41 4C
5	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	20 2B 15 40 20 4B 56 20 37 15 41 57 61 61 17 C
6	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	22 2D 38 43 22 2D 38 17 22 2D 38 4E 59 57 41
7	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	59 2D 38 43 22 2D 38 17 22 2D 38 4E 59 57 57 41
8	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	22 23 23 2D 17 41 21 2D 43 C 22 37 43 58 62 21
9	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	22 23 23 2D 17 17 C 20 22 C 22 C 38 43 43 43
10	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43	22 36 2B 4C 58 5A 15 20 62 41 2B 62 57 2B 20 2B

Pada Tabel 4.5 di bawah ini, dilakukan masukan Ki yang tetap hingga sepuluh percobaan yang dipasangkan dengan nilai RAND dengan data dinamis. Dan hal ini dilakukan dengan memasukkan nilai Ki dan RAND secara urut hingga sepuluh percobaan. Ki dan RAND yang dimasukkan sebagai data dipisahkan dengan tanda ” ; ” (tanpa tanda ”).

Pada Tabel 4.5 di bawah ini, dilakukan percobaan dengan menggunakan Ki yang tetap nilainya hingga sepulu kasli percobaan, sedangakan untu nilai RAND yang selalu berubah untuk sepuluh percobaan di bawah ini.

Tabel 4.5 Data Uji Coba Untuk Ki Tetap, RAND berubah.

No	Ki	RAND
1	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	12 12 23 45 65 76 87 98 43 43 54 65 76 87 98 32
2	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	34 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98
3	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	31 32 43 54 32 54 65 76 32 45 65 23 45 66 78 99
4	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	12 23 43 45 65 76 87 98 12 13 14 14 54 32 65 76
5	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	32 34 21 64 32 75 86 32 55 21 65 87 97 97 23 12
6	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	34 45 56 67 34 45 56 23 34 45 56 78 89 87 87 65
7	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	89 45 56 67 34 45 56 23 34 45 56 78 89 87 87 65
8	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	34 35 35 45 23 65 33 45 67 12 34 55 67 88 98 33
9	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	34 35 35 45 23 23 12 32 34 12 34 12 56 67 67 67
10	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67	34 54 43 76 88 90 21 32 98 65 43 98 87 43 32 43

Setelah dilakukan percobaan seperti tabel 4.5 tersebut di atas, maka dilakukan persamaan rumus aljabar algoritma A8, untuk menghasilkan *chipering key* Kc seperti pada Tabel 4.6 di bawah ini. Pada Tabel 4.6 di bawah ini, walaupun dilakukan masukkan Ki dan RAND sebesar 16 byte masing-masing dengan nilai yang berbeda-beda maupun tetap sama, maka diperlihatkan pada Tabel 4.6 di bawah bahwa jumlah

Kc tetap sama, sedangkan nilai urutan bit yang berbeda untuk tiap perubahan masukan yang berbeda.

Tabel 4.6 Hasil Kc (Ki Tetap dan RAND Berubah)

No	Kc
1	1110001001101111000111110011000011110010000011010110000000000000
2	0011101001110110101011010001110100101101111110110011110000000000
3	1000011010111011001110010001110110000100010010011111010000000000
4	0000101001010001100011001111100001100001000100101110010000000000
5	0101101001001100111001010011111011101000110000110010110000000000
6	0101011110001011101011010101100101111010111111001010100000000000
7	0011111100001111000101010010101001000101100100001101110000000000
8	0000010111101100110111100111100000111000011101100010000000000000
9	0001011000000110100100100001111110101001010010100010010000000000
10	0000111000101110001010101010001101010011011101000011100000000000

Sedangkan pada Tabel 4.7 di bawah ini, merupakan konversi untuk format byte ke hexadecimal. Dan sama pada percobaan sebelumnya, bahwa seluruh percobaan pada Tabel 4.7, Tabel 4,8 dan Tabel 4.9 dilakukan hal yang sama yaitu dengan menggunakan perhitungan aljabar algoritma A8.

Untuk keseluruhan tabel Ki dan RAND, dilakukan urutan-urutan secara bergantian antaranya untuk memperoleh array matrix hingga dua kolom. Dilakukan dua kolom karena untuk menghitung masukan yang berjumlah dua jenis yaitu Ki dan RAND.

Dari keseluruhan percobaan ini, menggunakan metode atau pola yang sama, yaitu kompresi dan permutasi. Ki dan RAND dapat berupa format byte, bit atau hexadesimal untuk kemudian dilakukan perhitungan dengan pola yang sudah ditetapkan dalam algoritma A8.

Hasil percobaan Ki dan RAND sudah pasti menghasilkan *chipering key* yang unik dan selalu berubah setiap kali data masukan yang berubah.

Seperti pada Tabel 4.7 di bawah ini, dilakukan dengan nilai Ki yang berubah, sedangkan RAND yang tetap.

DATA PERCOBAAN 3

Pada Tabel 4.7 di bawah, dilakukan sepuluh percobaan yang berbeda-beda untuk dapat dihasilkan Kc yang berbeda dan dapat dilakukan proses enkripsi.

Tabel 4.7 Konversi Hexadesimal ke Desimal

No.	Ki	RAND
1	22 C 17 2D 41 4C 57 62 2B 2B 36 41 4C 57 62 20	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
2	38 C 17 2D 41 4C 57 2D 2B 2B 36 41 4C 57 62 20	2D 2D 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
3	4E 2B 36 38 43 57 62 15 20 22 36 41 56 36 57 62	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
4	1F 20 2B 36 20 36 41 4C 20 2D 41 17 2D 42 4E 63	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
5	1F 20 2B 36 20 36 41 4C 20 2D 41 17 2D 42 4E 63	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
6	C C 17 2D 41 4C 57 62 2B 2B 36 41 4C 57 62 20	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
7	22 2B 36 38 43 57 62 15 20 22 36 41 56 36 57 62	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
8	62 C 17 2D 41 4C 57 62 43 2B 36 41 4C 57 62 20	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
9	22 2B 36 38 43 57 62 15 20 22 38 56 36 57 62	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43
10	20 2B 36 38 43 57 62 15 20 22 41 56 56 36 57 98	2D D2 22 2D 22 17 C 20 22 C 22 C 38 43 43 43

Dan dilakukannya percobaan hingga sepuluh kali, pada data Tabel 4.7 tersebut di atas, untuk mengetahui bahwa hasil akhir Kc yang selalu memiliki jumlah bit yang sama.

Pada Tabel 4.8 di bawah ini, dilakukan pengubahan data dari format hexadesimal ke dalam format byte atau desimal untu dapat dilakukannya komputasi data dan diperolehnya Kc dalam bentuk biner setelah dikonversi dari byte.

Tabel 4.8 Data Uji Coba Untuk Ki Berubah, RAND Tetap.

No.	Ki	RAND
1	34 12 23 45 65 76 87 98 43 43 54 65 76 87 98 32	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
2	56 12 23 45 65 76 87 45 43 43 54 65 76 87 98 32	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
3	78 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
4	31 32 43 54 32 54 65 76 32 45 65 23 45 66 78 99	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
5	31 32 43 54 32 54 65 76 32 45 65 23 45 66 78 99	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
6	12 12 23 45 65 76 87 98 43 43 54 65 76 87 98 32	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
7	34 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
8	98 12 23 45 65 76 87 98 67 43 54 65 76 87 98 32	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
9	34 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67
10	32 43 54 56 67 87 98 21 32 34 54 65 86 54 87 98	45 45 34 45 34 23 12 32 34 12 34 12 56 67 67 67

Dan pada Tabel 4.9 di bawah ini, diperlihatkan nilai Kc yang merupakan hasil dari nilai Ki yang tetap dan nilai RAND yang berubah. Dari data hasil percobaan yang didapatkan bahwa jumlah bit selalu sebanyak 64 bit, dan dengan nilai yang berbeda sesuai dengan masukan Ki dan RAND.

Dilakukannya substitusi terhadap Tabel GSM untuk memperoleh setiap *chipering key* Kc yang merupakan tahap permutasi dari hasil akhir kompres ke lima, seperti yang dilakukan pada Tabel 4.9 di bawah ini.

Tabel 4.9 Hasil Kc (Ki Berubah, RAND Tetap)

No	Kc
1	1111110111010000011100001000111000010000100100101111000000000000
2	0101111001000111010000101100100011100101110010010010110000000000
3	1010011000100001000110000000111001001110100100100011010000000000
4	1110000000001101011111101110000110000010111101111011000000000000

Tabel 4.9 Hasil Kc dengan Ki Berubah, RAND Tetap (lanjutan)

No	Kc
5	11100000000011010111111011100001100000101111011110110000000000000
6	11110010100010001110001011101101101000101110011010010100000000000
7	01001101111111101011110110110111010111001110011001001000000000000
8	00000111101110110100011101101111101010011111101100101000000000000
9	01001101111111101011110110110111010111001110011001001000000000000
10	01100010100010111101100001111111010100010010101000000100000000000

4.2.1.2. Uji karakteristik unjuk kerja proses pembangkitan Kc terhadap usaha intruder dalam menyadap informasi.

Para intruder berupaya untuk melakukan penyadapan informasi yang dikirim melalui jalur radio. Untuk itulah dibutuhkan pengamanan informasi menggunakan algoritma A5. Pada Tabel 4.10 di bawah ini, diperlihatkan uji coba enkripsi dengan *cipering key* sebagai pengacak informasi yang dikirim.

Tabel 4.10 Hasil Uji Coba Enkripsi

No	*Ciphering Key*	*Plaintext*	*Ciphertext*
1	11100010011011110001111100110000111100100 00011010110000000000000	Indonesia	3 -
2	10011101001111111011001000001011111001010 11001000000110000000000	guru	Pbeb
3	10010010111000100101001001111111000110111 01010000011100000000000	buku	
4	01011011010001110001101110010011010100100 11110010010110000000000	Kertas	R|kmxj
5	01101000100001000000111100111110001011001 10001000000010000000000	Laptop	s^OKPO
6	10111110111010100000101011111001000101011 10101011100000000000000	Komputer	rVTILM\K
7	00010110111100001010011000001000000111111 00011100001100000000000	Harddisk	#0&&+1)
8	01011110100110000101000100011000110011010 00000000100100000000000	Memory	/
9	01001101111111101011110110110111010111100 11100110010010000000000	UserManual	|ZL[dHG\HE
10	11000000101000101110110010100011010011110 11110101111100000000000	elektroniK	5

Tabel 4.10 Hasil Uji Coba Enkripsi (lanjutan)

No	*Ciphering Key*	*Plaintext*	*Ciphertext*
11	11100010011011110001111100110000111100100 00011010110000000000000	Teknik	.
12	00111010011101101010110100011101001011011 11110110011110000000000	Teknik	79<;9
13	10000110101110110011100100011101100001000 10010011111010000000000	Teknik	2
14	00001010010100011000110011111000011000010 00100101110010000000000	Teknik	j[UPWU
15	01011010010011001110010100111110111010001 10000110010110000000000	Teknik	[jdafd
16	01010111100010111010110101011001011110101 11111001010100000000000	Teknik	?1431
17	00111111000011110001010100101010010001011 00100001101110000000000	Kebaikan	\ruv~|vy
18	00000101111011001101111001111000001110000 11101100010000000000000	Kebaikan	@nijb`je
19	00010110000001101001001000011111101010010 10010100010010000000000	Kebaikan	vX_\TV\S
20	00001110001011100010101010100011010100110 11101000011100000000000	Kebaikan	S}zyqsyv
21	11111101110100000111000010001110000100001 00100101111000000000000	12345	,/.)(
22	01011110010001110100001011001000111001011 10010010010110000000000	54321	
23	10100110001000010001100000001110010011101 00100100011010000000000	Per3ADI	Xmz;ILA
24	11100000000011010111111011100001100000101 11101111011000000000000	2,345	~` xy
25	11100000000011010111111011100001100000101 11101111011000000000000	Apakah?	<-'-$s
26	11110010100010001110001011101101101000101 11001101001010000000000	!@#$%^&*()_+	J+HON5MACB4@
27	01001101111111101011110110110111010111100 11100110010010000000000	,./	
28	00000111101110110100011101101111101010011 11110110010100000000000	\M8*9	,=HZI
29	01001101111111101011110110110111010111100 11100110010010000000000	-12345	
30	01100010100010111101100001111111010100010 01010100000010000000000	+12345	}gdebc

Pada Tabel 4.10 tersebut di atas, ditunjukkan tiga pulu percobaan, di mana terdapat *ciphering key* yang telah berhasil dibangkitkan oleh algoritma A8 , dengan masukan berupa Ki dan RAND yang berbeda. Ki dan RAND yang digunakan dalam algoritma

A8 ini, adalah berkapasitas sebesar 128 bit. Lalu setelah dilakukan komputasi tersebut di atas, maka diperolehlah *ciphering key* dengan kapasitas 64 bit. Kunci *cipher key* ini digunakan sebagai masukan untuk algoritma A5, yang mana digunakan sebagai bit pengacak informasi yang telah dilakukan komputasi oleh algoritma A5 tersebut.

Pada percobaan satu hingga percobaan sepuluh dari Tabel 4.10 di atas, *ciphering key* yang dihasilkan merupakan hasil dari masukkan Ki dan RAND yang berbeda untuk setiap kali uji coba.

Setelah *ciphering key* berhasil dibangkitkan, maka setelah itu dilakukan proses enkripsi informasi. Artinya informasi yang akan dikirim atau *plaintext*, terlebih dahulu dikomputasikan dengan *ciphering key* tersebut di atas, lalu informasi yang telah diacak berupa *ciphertext* tersebut dikirim ke MS melalui proses hirarki interkoneksi. Ketika intruder melakukan penyadapan informasi melalui jalur radio, maka yang terbaca adalah berupa *ciphertext*.

Pada percobaan pertama, *plaintext* yang dikirim adalah ‘Indonesia’. Lalu setelah dilakukan komputasi terhadap *ciperhing key* , maka terbentuklah informasi acak atau *ciphertext* berupa ‘3 - ’ . Sedangkan pada percobaan kedua, *plaintext* yang dikirim adalah ‘guru’, lalu *ciphertext* yang dibangkitkan adalah ‘pbeb’. Kedua percobaan di atas, menunjukkan tingkat keamanan data yang baik, karena *ciphertext* yang dibangkitkan tidak sedekitpun dapat dipecahkan untuk mendapatkan informasi.

Sedangkan pada sepuluh percobaan kedua, digunakanlah informasi yang sama dengan *ciphering key* yang berbeda untuk setiap uji coba. *Plaintext* yang digunakan adalah kata ‘Teknik’. *Plaintext* tersebut sama untuk lima percobaan pertama. *Ciphering key yang digunakan pertama kali adalah* 1110001001101111000111110011000011111001000001101011000000000000, lalu *ciphering key* berikutnya adalah 00111010011101101010110100011101001011011111101100111100000000000, dengan *plaintext yang berbeda secara urut,* ’ . ’ dan ’ 79<;9’ . Hal ini menunjukkan bahwa walaupun digunakannya *plaintext* yang sama untuk setiap uji coba dengan menggunakan

cipheirng key yang bebeda , maka *ciperhtext* yang dibangkitkan akan selalu berbeda, sedangkan *plaintext* hasil dekripsi akan selalu sesuai dengan informasi sebelum dikirim.

Untuk sepuluh percobaan ketiga, digunakanlah *plaintext* berupa symbol, untuk mengetahui bahwa keamanan yang dilakukan tidak hanya berupa kata atau kalimat, namun juga dapat berupa *symbol*.

BAB 5

KESIMPULAN

Dari hasil uji coba simulasi proses pembangkitan kunci *ciphering key* 64 bit Kc, maka dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Ki dan RAND yang berbeda untuk setiap kali percobaan akan menghasilkan Kc yang berbeda nilainya, namun tetap jumlah bit-nya untuk setiap uji coba.
2. Kunci Kc yang sah bila digunakan untuk komputasi terhadap informasi yang akan dikirim, maka intruder tidak akan dapat membaca atau menyadapnya, karena telah dilakukan proses acak.
3. Proses acak ini dilakukan dengan perhitungan enkripsi menggunakan algoritma A5.
4. Bila Kc pada MS berbeda dengan Kc pada MSC, maka akan terjadi *error*. Hal ini menunjukan keamanan sistem yang terjamin.
5. Pada proses enkripsi, apabila hasil perhitungan *ciphering key* berbeda untuk setiap uji coba, dan dilakukan komputasi terhadap *plaintext* dengan menggunakan algoritma A5, maka hasilnya selalu memiliki nilai *ciphertext* yang berbeda.
6. Untuk setiap data informasi yang dilakukan secara acak menggunakan algoritma A5, maka data informasi tersebut akan aman.

DAFTAR ACUAN

[1] (2000). **“** Fraud dan Metode Sekuriti Pada Telekomunikasi Telepon Bergerak ”. Diakses 01 November 2008, dari Elektro Indonesia.
http://www.itb.ac.id

[2] Data sheet AT89S52

[3] Afgianto Eko Putra, *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*, Gava media, Edisi Kedua, Yogyakarta, 2004.

[4] *AT89S51/52 Development Tools DT-51 Minsys User Guide*, Innovative electronic.

[5] Barry B. Brey, *Mikroprocessors : 8086/ 8088, 80186/80188, 80286, 80386, 80486, Pentium, Prosesor Pentium Pro, Pentium II, Pentium III, dan Pentium 4 Edisi keenam*, Ed.I, ANDI, Yogyakarta: 2005

[6] JM Zacharias (2005). **“** Arsitekture GSM”. Diakses 01 Oktober 2008, http://www.jmzacharias.com/GSM.htm

[7] *Uke Kurniawan Usman, Ir, MT.* **“***Global Sistem for Mobile communication* (GSM)*”*, Diakses 01 Oktober 2008, dari STTTELKOM, Bandung.

http://www.stttelkom.ac.id

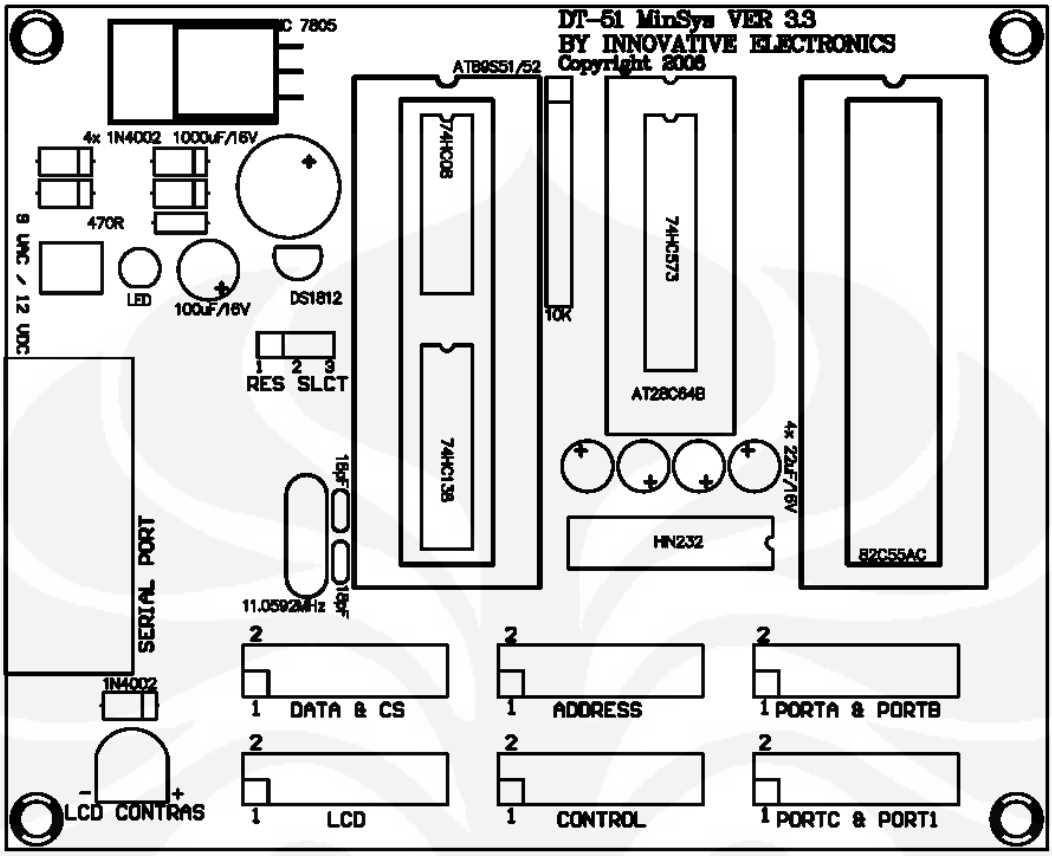
DAFTAR PUSTAKA

[1] Anonim, GSM Cloning, *http://www.isaac.cs.berkeley.edu/isaac/gsm -faq-html*, tanggal akses 20 November 2008

[2] Sudmeyer, Philipp (2006).*A performance oriented implementation of COMP128",* RuhrUniversity Bochum, Jerman.

[3] Sin, Susan. "COMP128." http://calliope.uwaterloo.ca/ssjsin/COMP128.pdf tanggal akses 20 November 2008

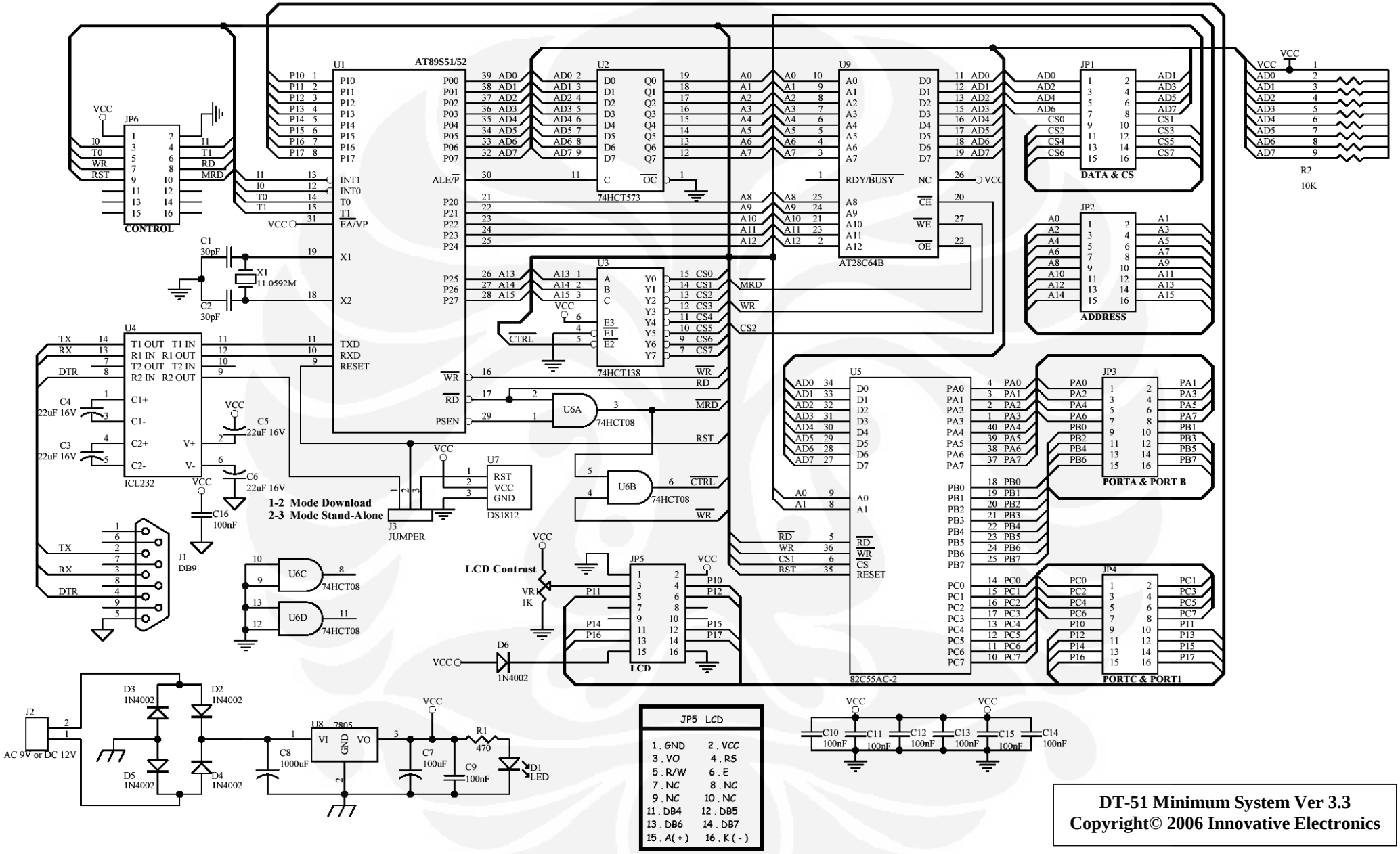
LAMPIRAN



Gambar 1-1
Tata Letak DT-51 Minimum System ver 3.3

Koneksi Kabel Serial DT-51 Minimum System ver 3.3

PC Serial Port Connector		DT-51 Minimum System ver 3.3 Serial Port Connector
DB9 Female	DB25 Female	DB9 Male
3	2	3
2	3	2
5	7	5
4	20	4


U1
AT89S51/52
U2
74HCT573
U3
74HCT138
U9
AT28C64B
U4
ICL232
U5
82C55AC-2
U6A
74HCT08
U6B
74HCT08
U6C
74HCT08
U6D
74HCT08
U7
DS1812
U8
7805
JP1
DATA & CS
JP2
ADDRESS
JP3
PORTA & PORT B
JP4
PORTC & PORT1
JP5
LCD
JP6
CONTROL
J1
DB9
J2
AC 9V or DC 12V
J3
JUMPER
X1
11.0592M
1-2 Mode Download
2-3 Mode Stand-Alone
LCD Contrast
VR1
1K
R1
470
R2
10K
D1
LED
D2
1N4002
D3
1N4002
D4
1N4002
D5
1N4002
D6
1N4002
JP5 LCD
1. GND
2. VCC
3. VO
4. RS
5. R/W
6. E
7. NC
8. NC
9. NC
10. NC
11. DB4
12. DB5
13. DB6
14. DB7
15. A(+)
16. K(-)
DT-51 Minimum System Ver 3.3
Copyright© 2006 Innovative Electronics

TABEL 32

```
  0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
 15, 12, 10,  4,  1 , 14, 11,  7,  5,  0,  14,  7,  1,  2, 13,  8,

  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
10 ,  3 , 4 , 9 , 6 , 0  , 3 , 2 , 5 , 6  , 8, 9 ,  11, 13, 15, 12
```

TABEL 64

```
 0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
1,  5,  29,  6,  25, 1,  18, 23, 17,  19, 0,  9,  24, 25,  6,  31,

 16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
28, 20,  24, 30, 4,  27,  3, 13,  15, 16, 14, 18,  4,  3,  8,  9,

 32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
20,  0, 12,  26, 21,  8, 28,  2, 29,  2, 15,  7,  11,  22, 14, 10,

 48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
17, 21, 12,  30, 26, 27, 16, 31,  11, 7,  13, 23, 10,  5,  22, 19
```

TABEL 128

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
52,	50,	44,	6,	21,	49,	41,	59,	39,	51,	25,	32,	51,	47,	52,	43,

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
37,	4,	40,	34,	61,	12,	28,	4,	58,	23,	8,	15,	12,	22,	9,	18,

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
55,	10,	33,	35,	50,	1,	43,	3,	57,	13,	62,	14,	7,	42,	44,	59,

48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
62,	57,	27,	6,	8,	31,	26,	54,	41,	22,	45,	20,	39,	3,	16,	56,

64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
48,	2,	21,	28,	36,	42,	60,	33,	34,	18,	0,	11,	24,	10,	17,	61,

80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
29,	14,	45,	26,	55,	46,	11,	17,	54,	46,	9,	24,	30,	60,	32,	0,

96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
20,	38,	2,	30,	58,	35,	1,	16,	56,	40,	23,	48,	13,	19,	19,	27,

112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
31,	53,	47,	38,	63,	15,	49,	5,	37,	53,	25,	36,	63,	29,	5,	7

TABEL 256

```
 0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
19, 11, 80, 114, 43, 1 , 69, 94, 39, 18, 127,1 17, 97,  3, 85, 43,

 16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
27, 124, 70, 83, 47, 71, 63, 10,  47, 89, 79,  4, 14, 59, 11,  5,

 32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
35, 107,103, 68, 21, 86, 36, 91, 85, 126, 32, 50,109, 94, 120,  6,

 48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
53, 79, 28,  45, 99, 95, 41,  34, 88, 68, 93, 55, 110,125,105, 20,

 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79
90, 80, 76, 96,  23, 60, 89, 64, 121, 56, 14, 74, 101, 8 , 19, 78,

 80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95
76, 66 ,104, 46,111, 50, 32,  3,  39,  0, 58, 25, 92,  22, 18, 51,

 96  97  98  99  100 101 102 103 104 105 106 107 108 109 110 111
57, 65, 119,116, 22, 109, 7 ,  86, 59,  93,  62,110,  78, 99,  77, 67,

112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
12, 113, 87,   98,102,  5,  88, 33,  38, 56,  23,  8,   75, 45,  13, 75,

128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
95,  63, 28,  49, 123, 120, 20, 112, 44, 30,  15,  98, 106, 2,  103, 29,

144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
82, 107, 42, 124, 24,  30,  41, 16, 108, 100, 117, 40, 73,  40,  7,  114,

160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
82, 115,  36, 112, 12,102, 100, 84,  92,  48, 72,  97,  9,  54, 55,  74,
```

```
 176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191
 113,123,  17,   26,  53,   58,   4,    9,   69,  122,  21,  118,  42,   60,   27,  73,

 192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207
 118,125,   34,  15,   65,  115,  84,   64,  62,  81,   70,   1,    24,111,  121,  83,

 208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223
 104,  81,  49,  127,  48,  105,   31,  10,   6,   91,   87,  37,   16,   54,  116,  126,

 224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239
  31,  38,   13,   0,   72,   106,  77,  61,  26,   67,   46,  29,   96,  37,   61,   52,

 240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255
101,  17,   44,   108,  71,  52,   66,  57,   33,   51,  25,   90,   2,    119,122,  35
```

TABEL AUTENTIKASI

TABEL 512

```
  0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15   →  Indeks ke
102,177 ,186,162, 2 ,156 ,112, 75, 55, 25, 8 , 12 ,251,193,246,188,  →  Nilai data kompresi

 16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
109 ,213,151, 53, 42, 79,191 ,115,233,242,164,223,209 ,148,108,161,

 32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
252 , 37,244, 47, 64,211 ,  6,237,185,160,139 ,113, 76,138, 59, 70,

 48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
67  , 26, 13,157, 63,179,221 , 30,214, 36,166 , 69,152,124,207,116,

 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79
247 ,194, 41, 84, 71,  1 , 49, 14, 95, 35,169 , 21, 96, 78,215,225,

 80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95
182 ,243, 28, 92,201,118 , 4, 74 ,248,128, 17, 11 ,146,132,245, 48,

 96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111
149 , 90,120, 39, 87 ,230,106 ,232,175 , 19 ,126,190 ,202 ,141 ,137,176,

112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
250 , 27 ,101, 40 ,219 ,227, 58 , 20, 51 ,178 , 98,216 ,140, 22 , 32 ,121,

128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
61 ,103 ,203 , 72 , 29,110 , 85,212 ,180 ,204 ,150,183 , 15, 66 ,172 ,196,

144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
56  ,197 ,158,  0,100 , 45 ,153 , 7 ,144 ,222 ,163,167,  60,135 ,210 ,231,

160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
```

```
174 ,165 , 38,249 ,224, 34 ,220 ,229,217 ,208 ,241, 68 ,206,189 ,125 ,255,

 176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191
239 , 54 ,168, 89,123 ,122 , 73 ,145 ,117,234,143 , 99 ,129,200 ,192, 82 ,

 192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207
104 ,170 ,136,235 , 93, 81 ,205 ,173,236 , 94 ,105, 52 , 46,228 ,198 , 5 ,

 208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223
57  ,254 , 97,155 ,142,133 ,199 ,171 ,187, 50 , 65,181 ,127,107 ,147 ,226,

 224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239
184 ,218 ,131 , 33, 77, 86 , 31 , 44, 88 , 62 ,238, 18 , 24, 43 ,154 , 23,

 240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255
80  ,159 ,134 ,111, 9 ,114 , 3 , 91  ,16 ,130, 83 , 10,195 ,240 ,253 ,119,

 256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271
177 ,102 ,162,186 ,156,  2 , 75 ,112, 25 , 55, 12 , 8 ,193 ,251 ,188 ,246,

 272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287
213 ,109 , 53 ,151, 79, 42 ,115 ,191 ,242,233 ,223 ,164,148,209 ,161 ,108,

 288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303
37 , 252 , 47,244 ,211, 64, 237,  6 ,160 ,185 ,113,139, 138, 76, 70,  59,

 304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319
26 ,  67 ,157, 13 ,179, 63 , 30 ,221, 36 ,214, 69 ,166 ,124,152 ,116 ,207,

 320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335
194 ,247 , 84, 41 , 1 , 71 , 14 , 49, 35, 95, 21  ,169, 78, 96  ,225 ,215,

 336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351
243 ,182 , 92, 28 ,118,201 , 74 , 4 ,128 ,248, 11 , 17 ,132 ,146 ,48 ,245,

 352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367
```

```
90  ,149 , 39,120 ,230, 87 ,232, 106, 19 ,175 ,190 ,126 ,141,202,176,137,

 368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383
27  ,250 , 40,101 ,227,219 , 20, 58 ,178 , 51,216 , 98 , 22 ,140,121, 32,

 384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399
103 , 61 , 72,203 ,110, 29 ,212 , 85,204 ,180 ,183,150 , 66, 15 ,196 ,172,

 400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415
197 , 56 , 0 ,158 , 45,100 , 7 ,153 ,222 ,144 ,167,163 ,135 , 60,231 ,210,

 416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431
165 ,174 ,249 , 38, 34 ,224,229 ,220 ,208,217 , 68 ,241,189,206 ,255 ,125,

 432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447
54  ,239 , 89,168 ,122,123 ,145 , 73,234 ,117 , 99,143 ,200,129 , 82, 192,

 448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463
170 ,104 ,235 ,136, 81, 93 ,173 ,205 , 94,236, 52 ,105,228 , 46 , 5 , 198,

 464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479
254 , 57 ,155 ,97 ,133,142 ,171 ,199, 50 ,187 ,181 ,65, 107,127 ,226 ,147,

 480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495
218 ,184,  33,131 , 86 , 77, 44 , 31, 62 , 88 , 18,238 , 43 ,24, 23 , 154,

 496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511
159  ,80 ,111,134 ,114, 9  , 91 , 3 ,130,  16 ,10 , 83 ,240 ,195,119 ,253
```