

**0806221311**

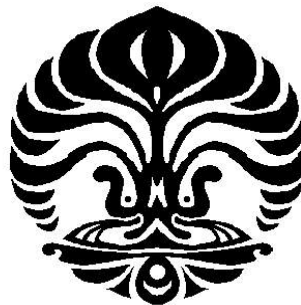
**Pemanfaatan Fasilitas GPRS Untuk Pengiriman  
Data KWH Meter Digital Menggunakan GSM Engine  
Dan Mikrokontroler**

**SKRIPSI**

diajukan sebagai persyaratan untuk  
menyelesaikan Pendidikan Sarjana program Instrumentasi Elektronika

**Farhan Nurdin**

**0305220267**



**Program Sarjana Ekstensi Fisika Instrumentasi**

**Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Indonesia**

**Depok**

**2008**

## LEMBAR PENGESAHAN

Judul Skripsi : **Pemanfaatan Fasilitas GPRS Untuk Pengiriman Data  
KWH Meter Digital Menggunakan GSM Engine Dan  
Mikrokontroler**

Nama : Farhan Nurdin

NPM : 0302220267

Jurusan : Ekstensi Fisika

Peminatan : Fisika Instrumentasi

Tanggal Sidang : 10 Juni 2008

Skripsi ini telah diperiksa dan disetujui oleh :

**Dosen Pembimbing**



**(Dr. rer.nat. Martarizal)**

**PENGUJI I**



**(Dr. Prawito)**

**PENGUJI II**



**(Drs. Arief Sudarmaji, M.T)**

## KATA PENGANTAR

Puji syukur yang teramat besar penulis ucapkan atas Rahmat dan Ridho yang telah Allah SWT berikan, yang telah memiliki kuasa atas semua yang ada di langit dan di bumi maupun yang berada di antara keduanya, sehingga skripsi ini bisa selesai tepat pada waktunya. Karena semua ini adalah atas kuasa dan keajaiban – MU. Ungkapan tersebut merupakan salah satu perwujudan syukur penulis kepada Allah SWT, semoga Allah SWT tetap menjaganya hingga keinginan tersebut benar – benar terwujud, amien.

Skripsi dengan judul **PEMANFAATAN FASILITAS GPRS UNTUK PENGIRIMAN DATA KWH METER DIGITAL MENGGUNAKAN GSM ENGINE DAN MIKROKONTROLER** dilakukan di Departemen Fisika FMIPA UI.

Selama penulis menyelesaikan kuliah dan menyusun skripsi ini, begitu banyak bantuan dan masukan yang sangat berarti sehingga penulis dapat menyelesaikan skripsi ini. Oleh karenanya pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Papa dan Mama yang telah memberikan kasih sayang, do'a dan perhatian yang tiada hentinya. Sari yang telah banyak membantu serta semua saudara-saudara yang telah memberikan motivasi untuk penulis.
2. Bapak Dr. rer.nat. Martarizal selaku dosen Pembimbing penulis serta pahlawan tanpa tanda jasa yang telah memberikan banyak bantuan, yang hingga kini penulis belum bisa membalas jasa baik Beliau.
3. Bapak Dr. Santoso Soekirno selaku Ketua Program Sarjana Ekstensi Fisika yang telah memberikan izin untuk dapat melaksanakan skripsi.
4. Mbak Lastri dan Mas Rizki, sahabat penulis yang juga banyak membantu dalam administrasi dan informasi kuliah.
5. Teman – teman penulis di Instrumentasi. Makasih semua bantuan dan perhatiannya selama kuliah

Dosen – dosen di jajaran Fisika, dan teman – teman Ekstensi Angkatan 2005 yang juga banyak membantu, serta semua pihak yang telah membantu penulis selama perkuliahan hingga menyusun skripsi ini, namun tidak dapat penulis sebutkan semua. Hanya Allah SWT yang dapat membalas semua kebaikan – kebaikan yang telah diberikan.

Penulis menyadari sepenuhnya bahwa masih banyak hal yang dapat menjadikan skripsi ini lebih baik, karenanya saran dan kritik yang bersifat membangun sangat diperlukan.

Depok, 31 Mei 2008

Penulis



## ABSTRAK

GPRS merupakan salah satu solusi untuk pengiriman data selain SMS, GPRS mampu terkoneksi ke jaringan internet dengan kecepatan maksimum 115,2 KBps, kecepatan ini tergantung padatnya jalur data pada BTS dan *coverege* dari jaringan GPRS, dengan tarif yang semakin murah dan terjangkau membuat GPRS sebagai solusi pengiriman data *alternatif*, pada skripsi ini GPRS difungsikan untuk mengirim dan menerima data dari halaman WEB, adapun protokol untuk pengiriman data menggunakan protokol FTP, sedangkan protokol yang digunakan untuk menerima data dari server adalah HTTP, FTP digunakan karena lebih efisien dan mudah mengaturnya dalam database.

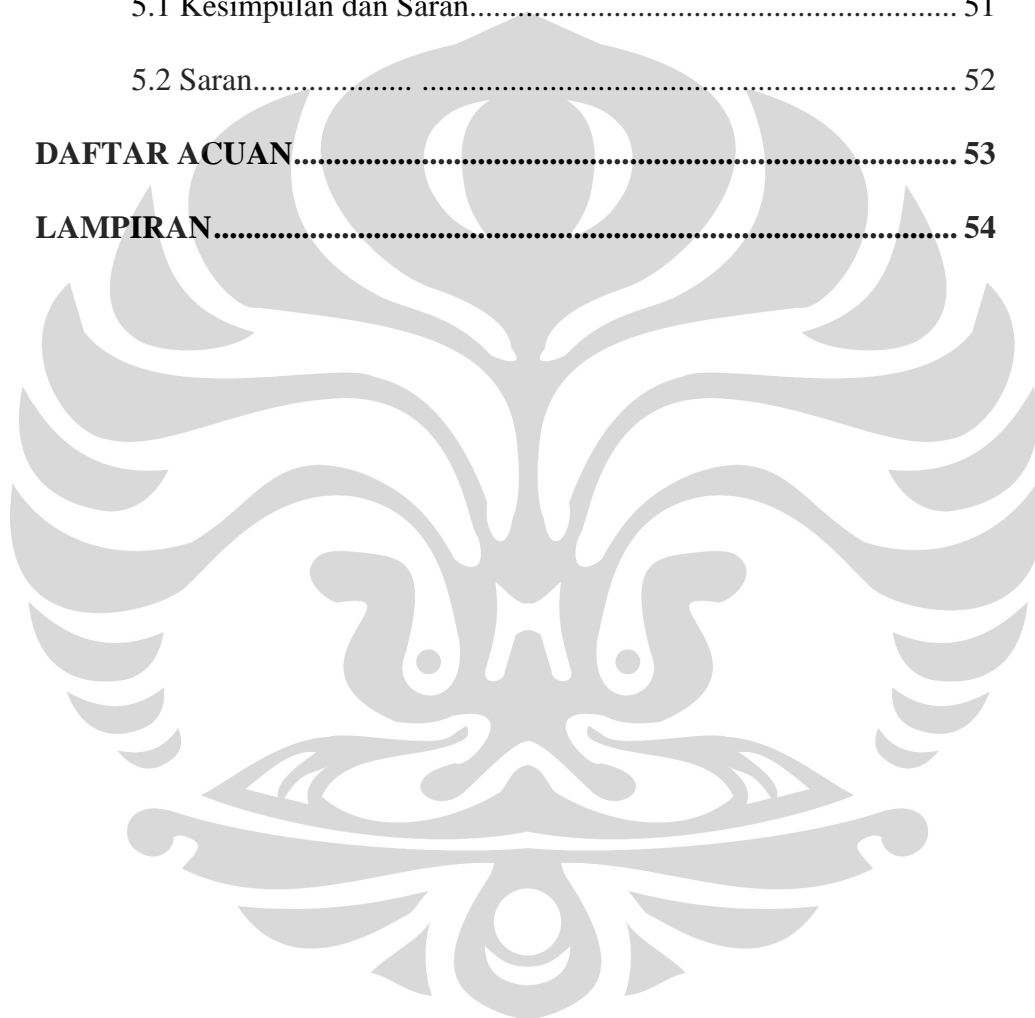
Kata kunci : WEB, HTTP, FTP, dan GPRS

## DAFTAR ISI

Lembar Pengesahan .....	ii
Kata Pengantar .....	iii
Abstrak .....	v
Daftar Isi .....	vi
Daftar Gambar.....	ix
Daftar Tabel .....	xi
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Penelitian .....	2
1.3 Pembatasan Masalah .....	2
1.4 Deskripsi Singkat .....	3
1.5 Metode Penelitian .....	4
1.6 Sistematika Penulisan .....	4
<b>BAB 2 TEORI DASAR.....</b>	<b>6</b>
2.1 GSM Engine MC55 .....	6
2.1.1 <i>GPRS</i> .....	6
2.1.1.1 <i>Protokol HTTP</i> .....	9
2.1.1.2 <i>Protokol FTP</i> .....	10
2.1.1.3 Socket Connections .....	12
2.2.1.4 Protokol SMTP .....	12

2.2.1.5 Protokol POP3.....	13
2.1.2 AT Command.....	13
2.1.3 Setting GPRS Untuk Setiap Operator .....	13
2.2 <i>Microcontroller AVR ATMEGA8535</i> .....	14
2.2.1 <i>Fitur dan Arsitektur AVR ATMEGA 8535</i> .....	15
2.2.2 Pengorganisasian Memori.....	16
2.2.2.1 <i>Program Memori</i> .....	16
2.2.2.2 <i>Data Memori</i> .....	17
2.2.3 <i>Konfigurasi Pin</i> .....	18
2.2.4 USART.....	19
2.3 <i>Komunikasi Data Serial</i> .....	21
2.3.2 <i>Komunikasi RS-232</i> .....	23
<b>BAB 3 PERANCANGAN DAN CARA KERJA SISTEM .....</b>	<b>24</b>
3.1 Blok Diagram dan Cara Kerja.....	24
3.2 GPRS Gateway.....	28
3.2.1 Service Initialization.....	29
3.2.2 Open Internet Service.....	35
3.2.3 Read / Write .....	36
3.3 Pengendali Mikro.....	37
3.4 Protokol Pengiriman Data.....	42
<b>BAB 4 DATA PENGUJIAN SISTEM.....</b>	<b>43</b>
4.1 Pengujian AT Command GSM Engine.....	43
4.2 Pengujian membuat Profil internet Connection .....	44

4.3 Pengujian membuat Profil internet Service Setup .....	46
4.4 Pengeksekusian Profil yang dibuat .....	47
4.4 Pengujian AT Command untuk mngetahui IP Address .....	49
<b>BAB 5 KESIMPULAN DAN SARAN.....</b>	<b>51</b>
5.1 Kesimpulan dan Saran.....	51
5.2 Saran.....	52
<b>DAFTAR ACUAN.....</b>	<b>53</b>
<b>LAMPIRAN.....</b>	<b>54</b>

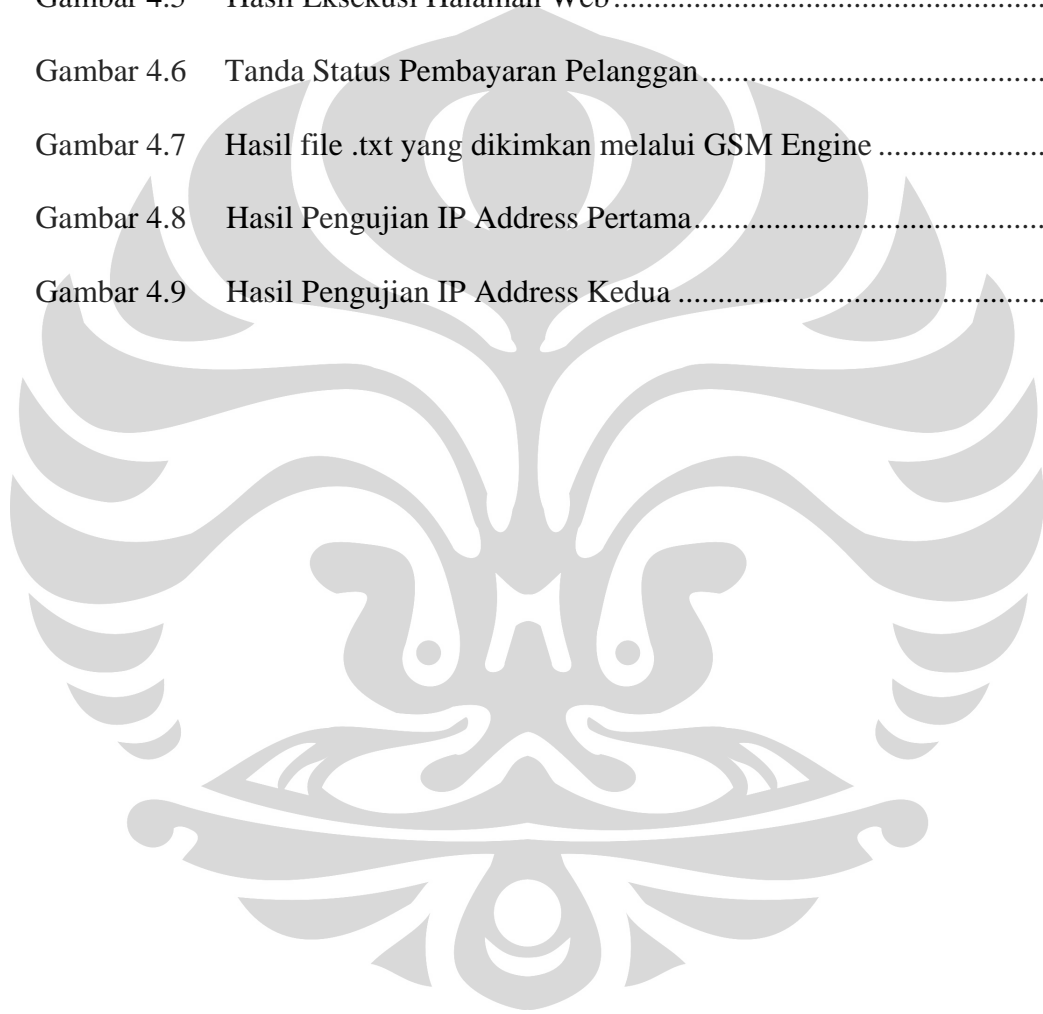




## DAFTAR GAMBAR

Gambar 1.1	Blok diagram Sistem Secara Keseluruhan .....	3
Gambar 2.1	Jaringan GPRS. ....	8
Gambar 2.2	Blok diagram mikrokontroler AVR ATmega8535.....	16
Gambar 2.3	Memori program dan memori data AVR ATMEGA 8535.....	17
Gambar 2.4	Konfigurasi pin ATMEGA 8535.....	19
Gambar 2.5	Blok Diagram USART .....	20
Gambar 2.6	Konfigurasi pin DB-9 <i>male</i> pada PC .....	22
Gambar 2.7	Konfigurasi pin DB-9 <i>female</i> .....	22
Gambar 2.8	Rangkaian sederhana RS-232.....	23
Gambar 3.1	Blok diagram perancangan sistem.....	24
Gambar 3.2	Flowchart sistem secara keseluruhan .....	26
Gambar 3.3	Blok Diagram Internet Service Via GPRS .....	28
Gambar 3.4	Diagram Pembuatan Profil FTP Get .....	29
Gambar 3.5	Diagram Pembuatan Profil FTP Put.....	30
Gambar 3.6	Diagram Pembuatan Profil SMTP.....	31
Gambar 3.7	Diagram Pembuatan Profil POP3.....	32
Gambar 3.8	Diagram Pembuatan Profil <i>Socket Connection</i> .....	33
Gambar 3.9	Diagram Pembuatan Profil HTTP.....	34
Gambar 3.10	Diagram <i>Open Internet Service</i> .....	35
Gambar 3.12	Diagram <i>Read</i> Data Menggunakan FTP Get.....	36
Gambar 3.13	Diagram <i>Write</i> Data Menggunakan FTP Put .....	37
Gambar 3.14	Rangkaian minimum sistem .....	38

Gambar 3.15	Rangkaian komunikasi serial antara mikro dengan GSM Engine.	38
Gambar 4.1	Data pengujian <i>AT command</i> GSM Engine pada Hyperterminal..	44
Gambar 4.2	Data pengujian <i>AT command Internet Connection Setup Profil...</i>	45
Gambar 4.3	Data pengujian <i>AT command Internet Service Setup</i> .....	46
Gambar 4.4	Data pengujian <i>AT Command</i> Untuk Mengeksekusi HTTP .....	47
Gambar 4.5	Hasil Eksekusi Halaman Web .....	47
Gambar 4.6	Tanda Status Pembayaran Pelanggan .....	48
Gambar 4.7	Hasil file .txt yang dikirimkan melalui GSM Engine .....	48
Gambar 4.8	Hasil Pengujian IP Address Pertama .....	49
Gambar 4.9	Hasil Pengujian IP Address Kedua .....	49



## DAFTAR TABEL

Tabel 2.1 AT Commands Untuk Inisialisasi GPRS.....	14
--	----



# BAB 1

## PENDAHULUAN

### 1.1 Latar belakang

Pertumbuhan telekomunikasi sekarang ini sangat berkembang dengan pesat, lebih dari 773 juta pengguna GSM diseluruh dunia berkembang setiap menitnya, penggunaan komunikasi ini tidak terbatas pada telepon, pengiriman pesan pendek atau multimedia, akses internet tetapi juga lainnya seperti mengakses content provider. GPRS sendiri hadir di Indonesia sekitar tahun 2001 lalu tetapi sampai sekarang penggunanya masih terbatas, hal ini disebabkan banyak faktor diantaranya masih sedikit orang mengenal kegunaan dari GPRS itu sendiri. GPRS atau *General Packet Radio Service* merupakan teknologi kedua transmisi data dari GSM atau teknologi 2,75G setelah *Circuit Switched Data* (CSD), kecepatan transmisi dari teknologi ini mencapai 115,2 kbps.

Keunggulan dari GPRS diantaranya ; memungkinkan seseorang mengakses ke jaringan perusahaan dan Internet, tarif untuk mengakses GPRS murah karena dihitung per kb atau kuota pemakaian, transmisi data yang cepat hingga 115,2 kbps, memungkinkan seseorang mengirim atau menerima data yang tidak dibatasi besarnya, dan lain-lain.

Pemanfaatan GPRS kali ini digunakan sebagai media transmisi data dari sebuah objek yang kita ukur dalam hal ini adalah KWH Meter ke sebuah penyimpanan data (database) melalui Alamat *Web* tertentu (*Web Server*), data yang dikirim tadi kemudian dapat diakses dimana saja melalui internet.

## 1.2 Tujuan Penelitian

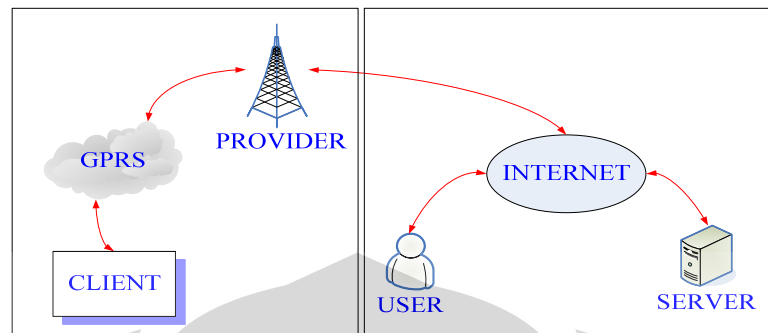
Skripsi merupakan suatu bentuk aktivitas mahasiswa yang dilakukan guna memenuhi salah satu mata kuliah di Universitas atau Perguruan Tinggi dan sebagai syarat untuk memperoleh gelar Sarjana. Sedangkan skripsi tersebut bertujuan :

- a. Untuk memahami dan memanfaatkan teknologi GPRS menjadi lebih baik.
- b. Dengan harapan sebagai hasil akhir, sistem yang digunakan dapat diaplikasikan untuk berbagai keperluan eksperimen, monitoring atau pengontrolan pada suatu proses dari berbagai aplikasi sesuai dengan yang hendak dipakai.
- c. Agar mahasiswa dapat mengetahui, dan mampu mengaplikasikan terhadap ilmu pengetahuan dan teori yang didapatkan dibangku perkuliahan dengan kondisi sebenarnya yang terjadi di lapangan, sehingga hasil dari aktivitas tersebut dapat membangkitkan mahasiswa lebih banyak membuka wawasan dalam bertindak atau berinovasi untuk pengembangan ilmu pengetahuan baik secara teori maupun praktek.

## 1.3 Batasan Masalah

Secara keseluruhan, sistem ini dibagi kedalam dua blok dimana tiap – tiap blok tersebut dikerjakan oleh mahasiswa yang berbeda, Blok-blok tersebut adalah blok Hubungan komunikasi hardware ke *web server* dan blok sitem *web server* dan database, Pada tugas akhir ini, penulis mengerjakan blok Hubungan komunikasi hardware ke *web server* yang berfungsi sebagai pengontrol dan pemantauan jarak jauh. Pembatasan masalah yang ditekankan adalah kemampuan

sistem untuk mengontrol perangkat elektronik dalam hal ini adalah KWH meter dengan menggunakan teknologi GPRS dan Mikrokontroler.



Gambar 1.1 Blok diagram sistem secara keseluruhan

#### 1.4 Deskripsi Singkat

Secara keseluruhan, sistem ini dibagi kedalam dua blok bagian dimana tiap-tiap blok tersebut dikerjakan oleh mahasiswa yang berbeda. Blok-blok tersebut adalah blok hardware yang terdiri atas mikrokontroler, KWH Meter Digital, dan GSM Engine, sedangkan blok yang lainnya merupakan software aplikasi berbasis Web Server. Pada blok hardware, mikro (client) berfungsi sebagai “otak” dari sistem tersebut yang mampu mengolah data masukan dan menghasilkan outputnya. Output yang dihasilkan oleh mikro selanjutnya ditransmisikan ke server untuk disimpan dalam bentuk database, blok diagram dapat dilihat pada gambar 1.1 diatas.

## 1.5 Metodologi Penelitian

Metode yang digunakan untuk pengerjaan dan penulisan Tugas Akhir antara lain:

- a. Studi literature: Mencari dan megumpulkan bahan-bahan yang diperlukan untuk menunjang penulisan laporan, pembuatan alat, dan referensi.
- b. Pembuatan alat dan pembuatan program: membuat hardware dari rangkaian elektronik yang digunakan dan membuat program mikrokontroler.
- c. Uji sistem: Uji sistem ini berkaitan dengan pengujian alat serta pengambilan data dari alat yang telah dibuat.
- d. Konsultasi: mengadakan konsultasi kepada dosen pembimbing tentang segala sesuatu yang berhubungan dengan tugas akhir

## 1.6 Sistematika Penulisan

Untuk memudahkan pembacaan dan pemahaman maka skripsi ini dibagi menjadi beberapa bab yaitu:

### **BAB 1   Pendahuluan**

Pendahuluan berisi latar belakang, permasalahan, batasan masalah, tujuan penulisan, metode penulisan dan sistematika penulisan dari skripsi ini.

### **BAB 2   Teori Dasar**

Teori dasar berisi landasan-landasan teori sebagai hasil dari studi literatur yang berhubungan dalam perancangan dan pembuatan alat (*hardware*) serta pembuatan program (*software*).

### **BAB 3 Perancangan Sistem**

Pada bab ini akan dijelaskan secara keseluruhan sistem kerja dari semua perangkat kontrol (*hardware*) dan program penghubung (*software*) yang terlibat antara mikrokontroler dengan KWH meter dan Mikrokontroler dengan GSM Engine.

### **BAB 4 Pengujian Sistem dan Pengambilan Data**

Bab ini berisi tentang cara kerja alat sebagai hasil dari perancangan sistem. Pengujian akhir dilakukan untuk memastikan bahwa sistem dapat berfungsi sesuai dengan baik. Setelah sistem berfungsi dengan baik maka dilanjutkan dengan pengambilan data untuk memastikan kapabilitas dari sistem yang dibangun.

### **BAB 5 Penutup**

Dalam bab ini berisikan kesimpulan akhir dari pengujian sistem dan saran untuk menyempurnakan alat dari tugas akhir ini.



## BAB 2

### TEORI DASAR

#### 2.1 GSM Engine MC55

GSM Engine MC55 merupakan produk keluaran dari Siemens, produk ini memiliki fitur antar lain <sup>[1]</sup> :

- a. Tri-Band GSM 900/1800/1900MHz
- b. SMS, Fax, CSD dan GPRS
- c. GPRS multi-slot class 10
- d. Control Via AT Commands
- e. SIM Applications Toolkit
- f. TCP IP Stack Via AT Commands
- g. Internet Service : TCP, UDP, HTTP, FTP, SMTP, POP3

##### 2.1.1 GPRS

GPRS (*General Packet Radio Service*) adalah suatu teknologi yang memungkinkan pengiriman dan penerimaan data lebih cepat jika dibandingkan dengan penggunaan teknologi *Circuit Switch Data* atau CSD. Sering disebut pula dengan teknologi 2,75G<sup>[2]</sup>.

GPRS merupakan sistem transmisi berbasis paket untuk GSM yang menggunakan prinsip “*tunnelling*”. Ia menawarkan laju data yang lebih tinggi. Laju datanya secara kasar sampai 160 kbps dibandingkan dengan 9,6kbps yang dapat disediakan oleh rangkaian tersakelar GSM. Kanal-kanal radio ganda dapat

dialokasikan bagi seorang pengguna dan kanal yang sama dapat pula digunakan secara berbagi (“*sharing*”) di antara beberapa pengguna sehingga menjadi sangat efisien.

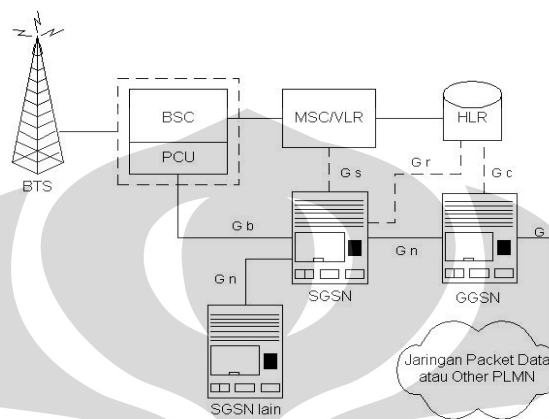
Dari segi biaya, pentarifan mengacu pada volume penggunaan. Penggunanya ditarik biaya dalam kaitannya dengan banyaknya byte yang dikirim atau diterima, tanpa memperdulikan panggilan, dengan demikian dimungkinkan GPRS akan menjadi lebih cenderung dipilih oleh pelanggan untuk mengaksesnya.

GPRS merupakan teknologi 2,75G dari GSM yang memungkinkan para operator jaringan komunikasi bergerak menawarkan layanan data dengan laju bit yang lebih tinggi dengan tarif rendah, sehingga membuat layanan data menjadi menarik bagi pasar massal. Para operator jaringan komunikasi bergerak di luar negeri kini melihat GPRS sebagai kunci untuk mengembangkan pasar komunikasi bergerak menjadi pesaing baru di lahan yang pernah menjadi milik jaringan kabel, yakni layanan internet. Kondisi ini dimungkinkan karena ledakan penggunaan internet melalui jaringan kabel (telepon) dapat pula dilakukan melalui jaringan bergerak. Sebagai gambaran kecil, layanan bergerak yang kini menjadi sukses di pasar (bagi operator di manca negara) misalnya adalah, laporan cuaca, pemesanan makanan, berita olah raga sampai ke informasi seperti berita-berita penting harian.

Perbedaan utama GPRS dan GSM adalah teknologi *packet switch* yang digunakan GPRS (*GSM circuit switch*), sehingga user menggunakan *resource* di *air interface* hanya pada saat ia akan mengirim atau menerima data saja. Pada saat user tidak mengirim atau menerima data, *timeslot* dapat digunakan oleh user lain. Dengan digunakannya *packet switch*, maka pentarif-an pada system GPRS dilakukan berdasarkan jumlah byte data yang *download*/*upload*nya (*volume*

*dependent charging*), bukan berdasarkan durasi waktu seperti pada system GSM (*time dependent charging*).

Gambar di bawah ini menunjukkan diagram jaringan sebuah system GPRS yang diupgrade dari system GSM [3].



Gambar 2.1 Jaringan GPRS

- a. SGSN (*Serving GPRS Support Node*) adalah *network equipment* yang berfungsi sama seperti MSC pada jaringan GSM. SGSN antara lain melakukan fungsi-fungsi *mobility management*, *routing/traffic management*, melakukan *authentikasi*, *chipering*, serta melakukan pemeriksaan IMEI.
- b. GGSN (*Gateway GPRS Support Node*) adalah *network equipment* GPRS yang berfungsi sama seperti Gateway MSC di jaringan GSM. GGSN merupakan interface antara jaringan GPRS dengan jaringan system lain seperti internet atau pun jaringan IP lainnya.
- c. PCU (*Packet Control Unit*) adalah *equipment* GPRS yang dipasang di BSC (sisi radio) yang antara lain melakukan fungsi-fungsi pengaturan kanal radio untuk GPRS seperti *power control* dan *congestion control*. Pada umumnya

PCU berfungsi sebagai *interface* antara jaringan *core* GPRS (SGSN dan GGSN) dengan BSS *sub system*.

Dalam teorinya GPRS menjanjikan kecepatan mulai dari 56 kbps sampai 115 kbps, sehingga memungkinkan akses internet, pengiriman data multimedia ke komputer, *notebook* dan *handheld computer*. Namun, dalam implementasinya, hal tersebut sangat tergantung faktor-faktor sebagai berikut:

- a. Konfigurasi dan alokasi time slot pada level BTS
- b. Software yang dipergunakan
- c. Dukungan fitur dan aplikasi ponsel yang digunakan

Ini menjelaskan mengapa pada saat-saat tertentu dan di lokasi tertentu akses GPRS terasa lambat, bahkan lebih lambat dari akses CSD yang memiliki kecepatan 9,6 kbps.

*Internet service* merupakan salah satu fitur yang terdapat pada GSM Engine MC55, fitur ini memanfaatkan teknologi GPRS sebagai barrier datanya, fitur – fitur tersebut antara lain TCP, UDP, HTTP, FTP, SMTP, POP3. Pada GSM Engine MC55 kita dapat membuat 10 profil sekaligus, adapun profil yang dapat dibuat untuk HTTP maksimum 2 profil, FTP hanya 1 profil, Socket Connections maksimum 6 profil, SMTP hanya 1 profil, dan POP3 hanya 1 profil<sup>[4]</sup>.

### **2.1.1.1 HTTP**

HTTP (*HyperText Transfer Protocol*) adalah protokol yang dipergunakan untuk mentransfer dokumen dalam *World Wide Web* (WWW). Protokol ini adalah protokol ringan, tidak berstatus dan generik yang dapat dipergunakan berbagai macam tipe dokumen.

Pengembangan HTTP dikoordinasi oleh *Konsorsium World Wide Web* (W3C) dan grup bekerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC 2616, yang menjelaskan HTTP/1.1, versi HTTP yang digunakan umum sekarang ini.

HTTP adalah sebuah protokol meminta/menjawab antara *client* dan *server*. Sebuah client HTTP seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya port 80). Sebuah server HTTP yang mendengarkan di port tersebut menunggu client mengirim kode permintaan (*request*), seperti “GET / HTTP/1.1” (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan dari data tertentu. Beberapa kepala (header) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban, seperti “200 OK”, dan sebuah pesan yang diminta, atau sebuah pesan *error* atau pesan lainnya.

#### 2.1.1.2 FTP

FTP (*File Transfer Protocol*) adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (file) komputer antar mesin-mesin dalam sebuah *internetwork*.

FTP merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan pengunduhan (*download*) dan penggugahan (*upload*) berkas-berkas komputer

antara klien FTP dan server FTP. Sebuah Klien FTP merupakan aplikasi yang dapat mengeluarkan perintah-perintah FTP ke sebuah server FTP, sementara server FTP adalah sebuah *Windows Service* atau *daemon* yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah klien FTP. Perintah-perintah FTP dapat digunakan untuk mengubah direktori, mengubah modus transfer antara biner dan ASCII, menggugah berkas komputer ke server FTP, serta mengunduh berkas dari server FTP.

Sebuah server FTP diakses dengan menggunakan *Universal Resource Identifier* (URI) dengan menggunakan format `ftp://namaserver`. Klien FTP dapat menghubungi server FTP dengan membuka URI tersebut.

FTP menggunakan *protokol Transmission Control Protocol* (TCP) untuk komunikasi data antara klien dan server, sehingga di antara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum transfer data dimulai. Sebelum membuat koneksi, port TCP nomor 21 di sisi server akan “mendengarkan” percobaan koneksi dari sebuah klien FTP dan kemudian akan digunakan sebagai port pengatur (*control port*) untuk (1) membuat sebuah koneksi antara klien dan *server*, (2) untuk mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada *server* dan juga (3) mengembalikan *respons server* ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka server akan mulai membuka port TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mentransfer data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan penggugahan.

FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi. Pengguna

terdaftar dapat menggunakan username dan password-nya untuk mengakses, *download*, dan meng-*upload* berkas-berkas yang ia kehendaki. Umumnya, para pengguna terdaftar memiliki akses penuh terhadap beberapa direktori, sehingga mereka dapat membuat berkas, membuat direktori, dan bahkan menghapus berkas. Pengguna yang belum terdaftar dapat juga menggunakan metode *anonymous login*, yakni dengan menggunakan nama pengguna *anonymous* dan *password* yang diisi dengan menggunakan alamat *e-mail*.

### **2.1.1.3 Socket Connections**

UDP, singkatan dari *User Datagram Protocol*, adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara host-host dalam jaringan yang menggunakan TCP/IP.

### **2.1.1.4 SMTP**

SMTP (*Simple Mail Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk pengiriman surat elektronik di Internet. Protokol ini dipergunakan untuk mengirimkan data dari komputer pengirim surat elektronik ke server surat elektronik penerima.

Protokol ini timbul karena desain sistem surat elektronik yang mengharuskan adanya server surat elektronik yang menampung sementara sampai surat elektronik diambil oleh penerima yang berhak.

### 2.1.1.5 POP3

POP3 (*Post Office Protocol version 3*) adalah protokol yang digunakan untuk mengambil surat elektronik (email) dari server email.

Protokol ini erat hubungannya dengan protokol SMTP dimana protokol SMTP berguna untuk mengirim surat elektronik dari komputer pengirim ke server.

Protokol POP3 dibuat karena desain dari sistem surat elektronik yang mengharuskan adanya server surat elektronik yang menampung surat elektronik untuk sementara sampai surat elektronik tersebut diambil oleh penerima yang berhak. Kehadiran server surat elektronik ini disebabkan kenyataan hanya sebagian kecil dari komputer penerima surat elektronik yang terus-menerus melakukan koneksi ke jaringan internet.

### 2.1.2 AT Commands

*AT Command* untuk tiap *device* berbeda-beda tapi pada dasarnya sama. Dalam perancangan alat akan digunakan *AT Command* yang *compatible* dengan GSM Engine MC55 Siemens. *AT Command* mirip dengan > (*Prompt*) pada DOS. Semua perintah harus diawali dengan kata *AT*.

### 2.1.3 Setting GPRS

Untuk dapat menggunakan GPRS, Sim Card GSM harus terlebih dahulu melakukan aktivasi GPRS,

Setting GPRS pada setiap kartu berbeda, berikut merupakan setting GPRS untuk GSM Engine pada setiap operator :



Tabel 2.1 AT Commands Untuk Inisialisasi GPRS

Operator	AT Commands
	AT^SICS=1,conType,GPRS0
	AT^SICS=1,alphabet,1
telkomsel	AT^SICS=1,user,wap
	AT^SICS=1,passwd,wap123
	AT^SICS=1,apn,www.wap.telkomsel.com
xl	AT^SICS=1,user,xlgprs
	AT^SICS=1,passwd,proxl
	AT^SICS=1,apn,www.wap.lifeinhand.com
IM3	AT^SICS=1,user,gprs
	AT^SICS=1,passwd,im3
	AT^SICS=1,apn,www.wap.indosat-m3.net
Mentari	AT^SICS=1,user,<blank>
	AT^SICS=1,passwd,<blank>
	AT^SICS=1,apn,www.satelindogprs.com

## 2.2 *Microcontroller* AVR ATMEGA 8535

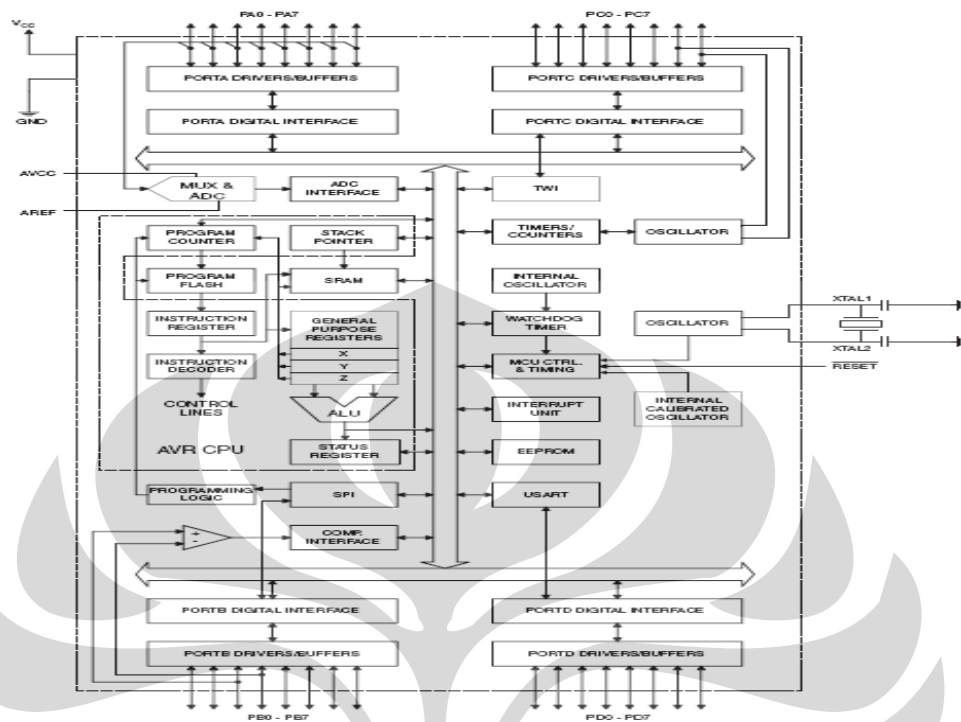
ATmega8535 adalah mikrokontroler CMOS 8-bit berdaya rendah yang memiliki arsitektur AVR RISC (*Reduced Instruction Set Computer*). Setiap instruksi mikrokontroler ATmega8535 dapat dieksekusi dalam satu siklus clock osilator, dan keluarannya bisa mencapai hampir sekitar 1 *MIPS* (*Million Instruction Per Second*) per MHz, sehingga konsumsi daya bisa optimal dan kecepatan proses eksekusi menjadi maksimal <sup>[5]</sup>.

### 2.2.1 Fitur Dan Arsitektur AVR ATMEGA 8535

Berikut ini alah fitur-fitur utama yang dimiliki oleh mikrokontroler Atmega8535:

- a. 8 KByte untuk *System Programmable Flash* pada *Read- While- Write* (ROM)
- b. Memori data EEPROM 512 Byte
- c. Memori data SRAM 512 Byte
- d. 32 Jalur Input-output
- e. 32 Register umum
- f. 3 *Flexible Timer / counter* dengan *Compare-Modes*
- g. Internal dan eksternal *Interrupt*
- h. Pemrograman serial dengan USART
- i. Interface serial Two-wire sebesar 1Byte
- j. 8 Single Ended Channels, 10 bit untuk ADC dan 4 kanal PWM
- k. Watchdog timer yang dapat deprogram dengan Internal Oscillator
- l. Sebuah serial port SPI
- m. 6 pilihan software dengan power save mode yaitu Idle, ADC Noise Reduction, Power save, Power down, Standby dan Extended Standby
- n. Memiliki kecepatan 0-8 MHz, dan beroperasi pada tegangan 2.7-5.5

Berikut adalah gambar arsitektur dari mikrokontroler Atmega 8535:



Gambar 2.2 Blok diagram mikrokontroler AVR ATmega8535

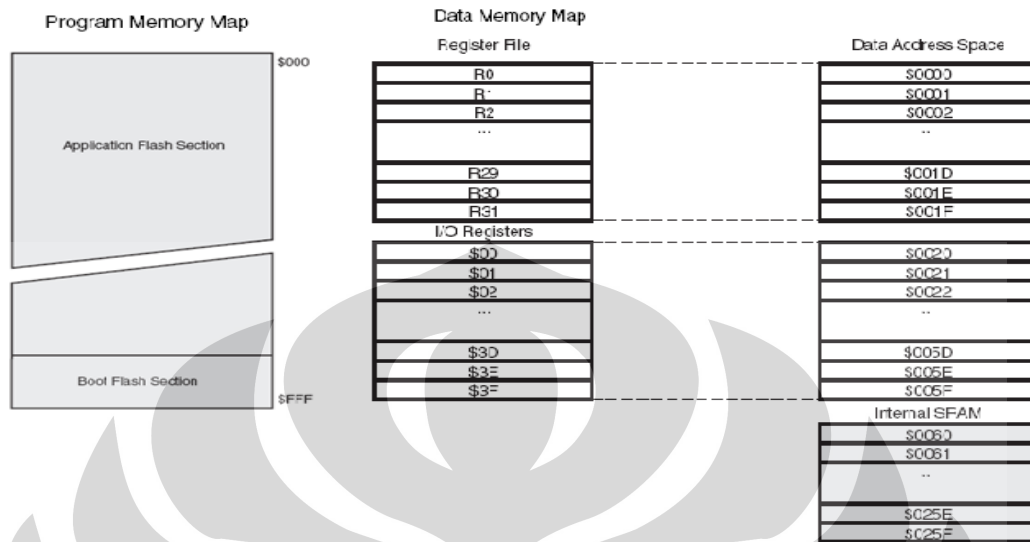
## 2.2.2 Pengorganisasian Memori

Mikrokontroler AVR Atmega8535 memiliki dua ruang utama memori yaitu memori data dan memori program. Sebagai tambahan terdapat memori EEPROM untuk menyimpan data. Berikut ini akan dijelaskan satu-per satu mengenai ketiga media penyimpanan tersebut:

### 2.2.2.1 Program Memory

Atmega 8535 terdiri dari 8 Kbyte On-Chip In-system Reprogrammable Flash Memory untuk penyimpanan program. Selama besar instruksinya adalah 16 sampai 32 bit, Flash akan terorganisasi sebesar 4K x 16 <sup>[2]</sup>. Untuk keamanan

software, memori program *Flash* dibagi menjadi dua bagian yaitu Program Boot dan Program Aplikasi.



Gambar 2.3 Memori program dan memori data AVR ATMEGA 8535

Memori Flash memiliki kemampuan paling tidak 10.000 kali dihapus dan ditulis. Program Counter-nya (PC) sebesar 12 bit, terletak pada alamat 4K pada memori program.

#### 2.2.2.2 Data Memory

Pada gambar 2.3 ditunjukkan bagaimana SRAM memori pada ATMEGA8535 diorganisasikan. 608 lokasi alamat data memori berisi register file, I/O memori, dan internal data SRAM. 96 lokasi alamat perlama berisi register file, dan I/O memori dan selanjutnya 512 lokasi alamat dari internal data SRAM. Terdapat lima mode pengalamatan yang berbeda pada data memori, yaitu: mode pengalamatan langsung, tidak langsung dengan displacement, tidak langsung, tidak langsung dengan pre-decrement, dan tidak langsung dengan post-

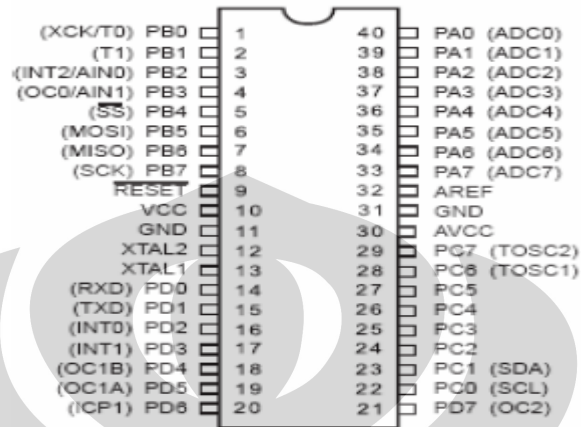
increment. 32 register fungsi umum, 64 I/O register, dan 512 byte internal data SRAM pada ATMEGA8535 secara keseluruhan dapat diakses menggunakan semua mode pengalamatan . Peta dari data memori dapat dilihat pada gambar 2.3

### 2.2.3 Konfigurasi Pin

Mikrokontroler AVR Atmega 8535 memiliki 40 pin yang terdiri dari :

- a. PORT A(A0-A7): Port A selain sebagai port I/O 8-bit bi-directional, juga melayani input analog sebagai A/D Converter.
- b. PORT B (B0-B7):Port B merupakan port I/O 8-bit bi-directional, selain itu port B juga mempunyai beberapa fungsi tambahan.
- c. PORT C (C0-C7): merupakan port I/O 8-bit bi-directional, selain itu port C juga mempunyai fungsi alternatif.
- d. PORT D (D0-D7): merupakan port I/O 8-bit bi-directional, selain itu port D juga mempunyai fungsi alternatif.
- e. VCC: Power Supply
- f. GND : Ground
- g. RESET: Reset input. Kondisi low berlangsung lebih panjang dari panjang pulsa minimum akan membuat mikrokontroler berada pada kondisi reset. Pulsa yang lebih pendek tidak menjamin timbulnya kondisi reset.
- h. AVCC : AVCC adalah tegangan supply untuk input port analog (Port A) dan AID *converter*. Saat ADC tidak digunakan AVCC terhubung langsung ke VCC, dan pada saat ADC digunakan AVCC akan terhubung ke VCC yang melalui *low pass filter*
- i. AREF: AREF adalah referensi analog untuk AID *converter*

- j. XTAL1: Input untuk *inverting oscillator amplifier* dan input untuk clock internal.
- k. XTAL2: Output *inverting oscillator amplifier*.



Gambar 2.4 Konfigurasi pin ATMEGA 8535

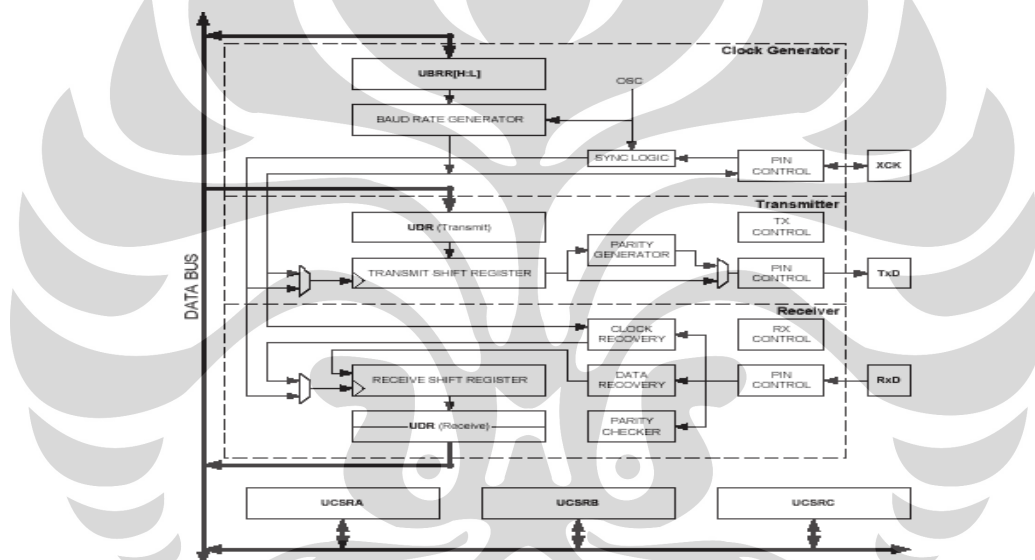
#### 2.2.4 *Universal Synchronous Asynchronous Receiver Transmitter (USART)*

##### *Universal Synchronous and Asynchronous Receiver Transmitter (USART)*

piranti komunikasi serial dengan fleksibilitas yang tinggi. Fasilitas ini berguna agar sistem aplikasi AVR dapat berkomunikasi dengan system lainnya. Fitur-fitur yang dimiliki USART adalah:

- a. Operation *Full duplex* ( memiliki register serial Rx dan Tx bebas)
- b. Operasi sinkron dan asinkron
- c. Operasi *Master or Slave Clocked Synchronous*
- d. Generator berkecepatan dan beresolusi tinggi
- e. Didukung *Serial Frame* dengan 5.6.7.8. atau 9 bit data dan 1 atau 2 *stop* bit
- f. Generasi paritas ganjil atau genap dan pengecekan paritas yang didukung oleh hardware

- g. Deteksi data yang berlebih
- h. Deteksi kesalahan *frame*
- i. Penyaringan *noise* termasuk deteksi kesalahan bit pertama dan digital *low pass filter*
- j. Memiliki tiga bu interrupt yang terpisah. Tx *Complete*, Tx *Data Register Empty*, dan Rx *Complete*
- k. Mode komunikasi *Multi-Processor*
- l. Mode komunikasi *Asynchronous* dengan kecepatan ganda.



Gambar 2.5 Blok Diagram USART

USART harus terlebih dahulu diinisialisasi, tahap penginisialisasian terdiri atas *baud rate*, *Setting Frame Format*, meng-enable-kan *Transmitter* dan *Receiver* sesuai penggunaan. Untuk pemakaian interrupt pada USART pada saat inisialisasi *Global Interrupt Flag* di fungsikan.

Pemakaian *Transmitter* USART, dengan cara mengatur *Transmit Enable* pada register UCSRB. Setelah itu operasi normal port pada, pin TxD ditolak oleh

USART dan akan difungsikan sebagai *Transmitter's Serial Output*. Pada saat operasi sinkron terjadi *Clock* dari pin XCK tidak digunakan sebagai gantinya digunakan *Clock* pengiriman (transmisi).

Pemakaian *Receiver* USART, dengan cara mengatur *Receive Enable* pada register UCSRB Setelah itu operasi normal port pada pin RxD ditolak oleh USART dan akan difungsikan sebagai *Receiver's Serial Input*. Pada saat operasi sinkron terjadi *Clock* dari pin XCK digunakan sebagai *Transfer Clock*.

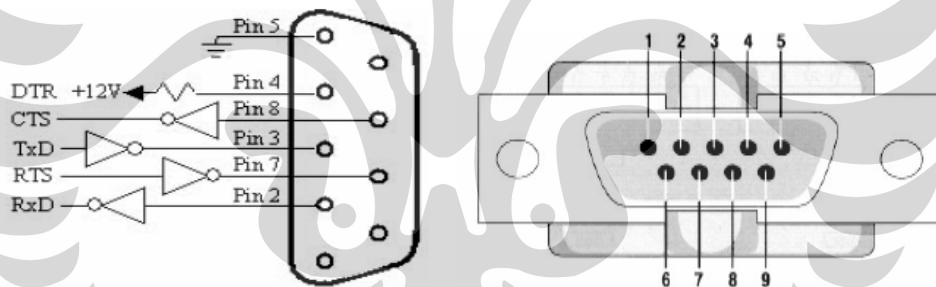
### 2.3 Komunikasi Data Serial

Komunikasi serial adalah pengiriman data secara serial (data dikirim satu persatu secara berurutan) sehingga komunikasi serial jauh lebih lambat daripada komunikasi yang paralel. Karena peralatan berkomunikasi menggunakan transmisi serial sedangkan data dikomputer diolah secara paralel, oleh karena itu harus dikonversikan dahulu ke bentuk paralel. Jika menggunakan perangkat keras hal ini bisa dilakukan oleh *Universal Asynchronous Receiver Transmitter (UART)*, yang membutuhkan perangkat lunak untuk menanganinya. Komunikasi serial merupakan salah satu cara untuk mengkomunikasikan data dari suatu peralatan ke peralatan lain dengan cara menggunakan data secara serial, misalnya mengkomunikasikan antara GSM Engine dengan Mikrokontroler, HP dengan PC, *printer* dengan PC dll Pada PC, komunikasi serial RS232 dapat dilakukan melalui *port* serial ( *COM port* ). Komunikasi data serial dapat dilakukan dengan mempresentasikan data dalam bentuk *level* "1" atau "0". Kelebihan komunikasi serial adalah jangkauan panjang kabel yang lebih jauh dibanding paralel karena serial *port* mengirimkan logika 1 dengan kisaran tegangan -3 Volt hingga -25 Volt

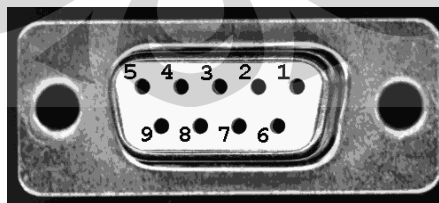


dan logika nol sebagai +3 Volt hingga +25 Volt sehingga kehilangan daya karena panjang kabel bukan masalah utama. Selain itu juga komunikasi serial *port* bersifat asinkron sehingga sinyal detak tidak dikirim bersama data. Setiap *word* disinkronkan dengan *start* bit dan sebuah *clock internal* di kedua sisi menjaga bagian data saat pewaktuan (*timing*).

Perangkat keras pada komunikasi serial dibagi menjadi dua kelompok, yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Contoh DCE adalah modem, sedangkan contoh DTE adalah terminal di komputer. *Null modem* digunakan untuk menghubungkan dua buah data DTE. Hal ini biasanya dilakukan untuk mengirim file antar komputer .



Gambar 2.6 Konfigurasi pin DB-9 *male* pada PC



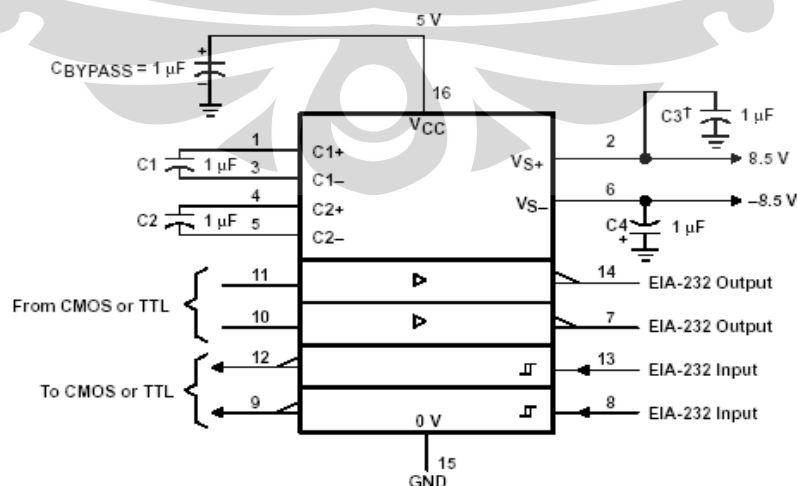
Gambar 2.7 Konfigurasi pin DB-9 *female*

### 2.3.1 Komunikasi RS-232

Standar sinyal komunikasi serial yang banyak digunakan adalah standar RS-232 yang dikembangkan oleh (*Electronic Industry Association and Telecommunications Industry Association EIA/TIA*). Standar ini hanya menyangkut komunikasi data antara komputer (*Data Terminal Equipment*) DTE dengan alat-alat pelengkap komputer DCE (*Data Circuit Terminating Equipment*), standar inilah yang biasa digunakan pada port serial IBM PC kompatibel [6].

Komunikasi serial RS232 merupakan komunikasi *asynchronous* sehingga sinyal *clock* tidak dikirim bersamaan dengan data. Setiap data disinkronisasikan dengan menggunakan *start bit* dan *clock* internal pada setiap *bit*. Port RS232 pada komputer harus memenuhi standar RS232. Agar *level* tegangan sesuai dengan tegangan TTL/CMOS diperlukan RS232 *level* konverter.

Apabila peralatan yang digunakan menggunakan TTL, sinyal serial *port* harus dikonversikan dahulu ke pulsa TTL sebelum digunakan. Sebaliknya, sinyal dari peralatan harus dirubah ke logika RS-232 sebelum dimasukkan ke serial *port*.



Gambar 2.8 Rangkaian sederhana RS-232

## BAB 3

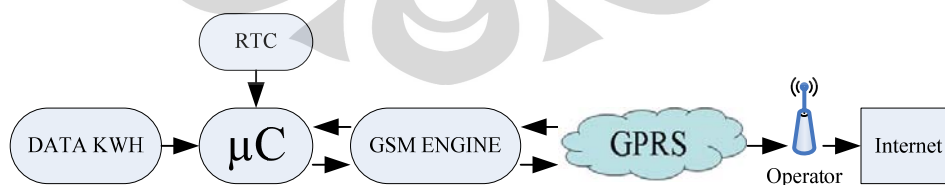
### PERANCANGAN DAN CARA KERJA SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja dari masing-masing hardware serta software yang digunakan penulis dalam pembuatan alat “Pemanfaatan Fasilitas GPRS Untuk Pengiriman Data KWH Meter Digital Menggunakan GSM Engine Dan Mikrokontroler”.

#### 3.1 Blok Diagram dan Cara Kerja

Dalam bab ini selain perancangan alat, juga akan dibahas mengenai cara kerja alat. Pada dasarnya sistem ini terdiri dari dua blok (seperti gambar 1.1) dimana masing-masing blok memiliki fungsi yang berbeda-beda, dan jika disatukan menjadi satu kesatuan masing-masing blok tersebut akan menjadi sebuah sistem yang terintegrasi.

Pada pembahasan ini akan dibahas cara kerja dari sistem dari blok hardware yang terdiri atas mikrokontroler dan GSM Engine. Diagram blok dari rancangan alat ini dapat terlihat pada gambar 3.1. sebagai berikut :



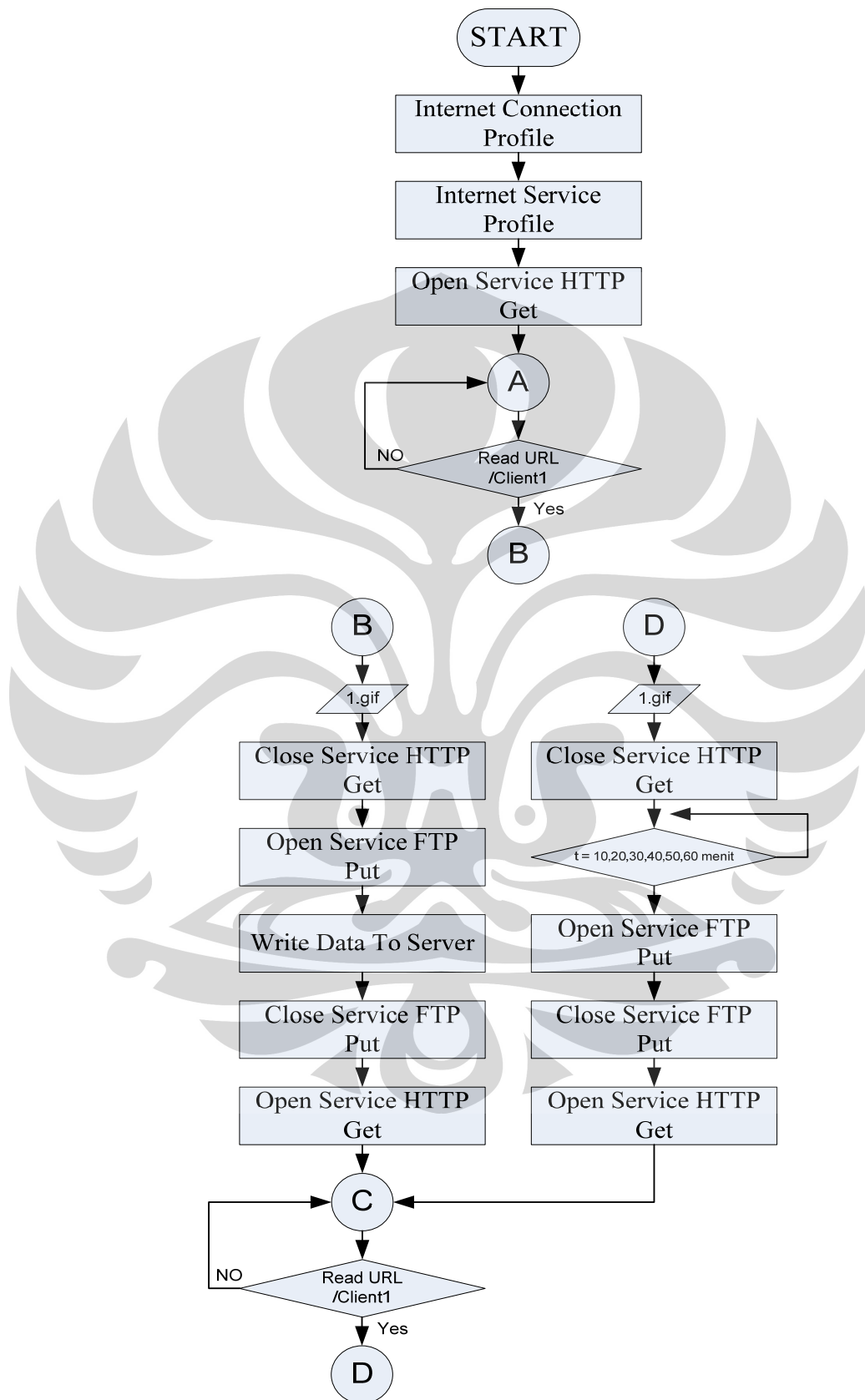
Gambar 3.1 Blok diagram perancangan sistem

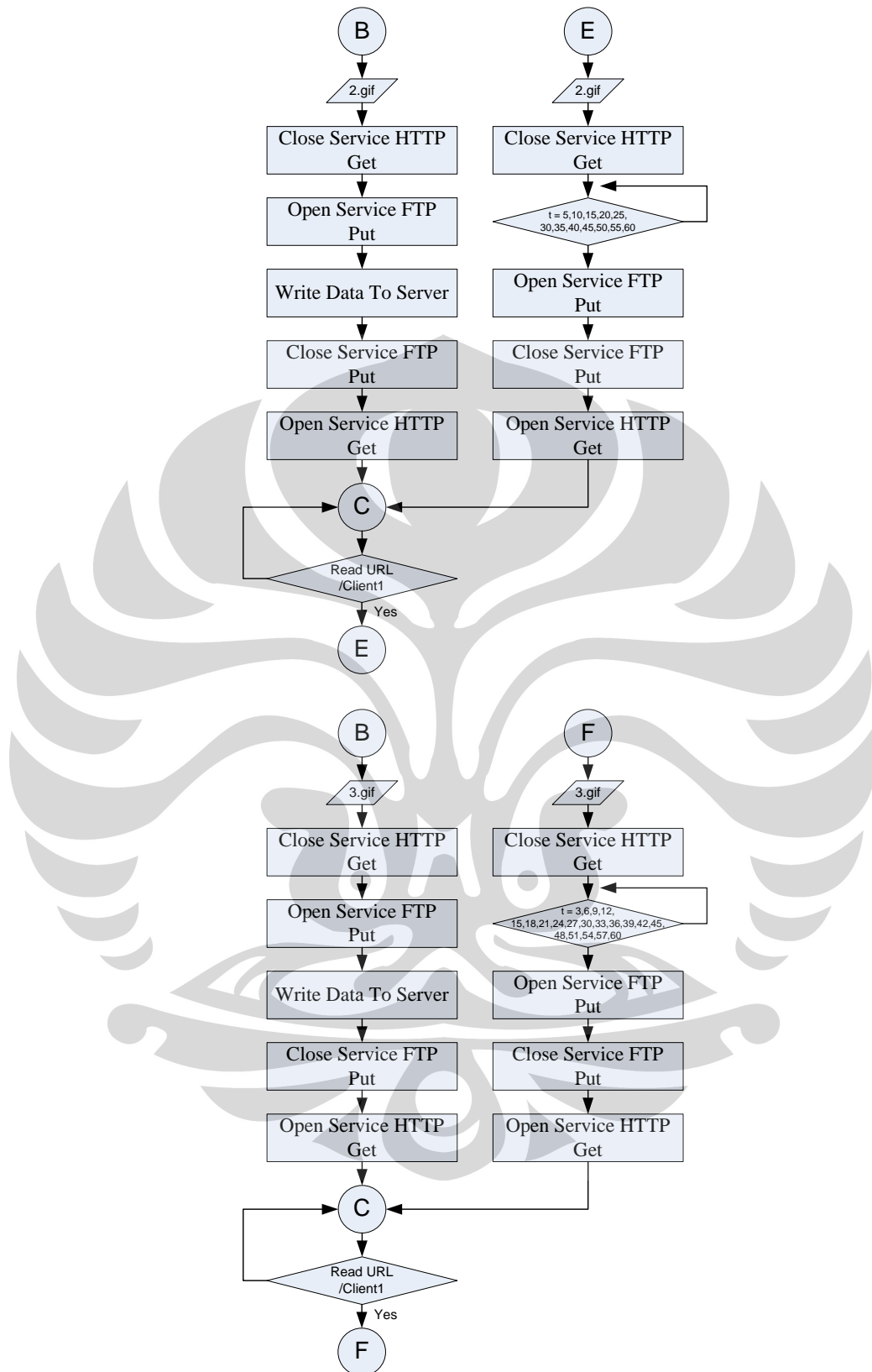
Deskripsi singkat dari blok diagram diatas memperlihatkan suatu sistem komunikasi yang menghubungkan dari satu *device* dengan *device* lainnya, dimana sistem komunikasinya berbasis komunikasi serial (kecuali dari GSM Engine ke GPRS *Gateway*). Komunikasi serial yang digunakan disini menggunakan RS-232. Cara kerja dari sistem ini adalah sistem ini terdiri dari beberapa device yang memiliki peranan penting pada masing-masing tugasnya, dimana dari blok diagram diatas dapat kita lihat bagian-bagian dari peran *device-device* tersebut, dimana GPRS *gateway* akan menerima data dari web server, yang kemudian data tersebut diterjemahkan oleh mikrokontroler untuk dieksekusi. Secara *default* pada halaman web /client1 terdapat sebuah file .gif, file tersebut dapat berarti apakah mikro kemudian mengirim data KWH setiap 10 menit untuk status pembayaran "lunas", *implementasi* dari status ini adalah mikro melakukan pengiriman data setiap bulan.

Sistem kerja GPRS Gateway yang digunakan adalah, jika pada halaman web tidak terdapat file 1.gif maka mikrokontroler memerintahkan GSM Engine untuk file ke server setiap 10 menit sekali, apabila pada halaman web terdapat file 2.gif maka mikrokontroler maka mikrokontroler memerintahkan GSM Engine untuk file ke server setiap 5 menit sekali, dan apabila pada halaman web terdapat file 3.gif maka mikrokontroler maka mikrokontroler memerintahkan GSM Engine untuk file ke server setiap 3 menit sekali,

File .gif tersebut juga menerangkan status pembayaran dari pelanggan, untuk 1.gif berarti "lunas", 2.gif berarti "belum bayar selama satu bulan", dan 3.gif berarti belum bayar lebih dari dua bulan.

## FLOWCHART CARA KERJA SISTEM GPRS GATEWAY

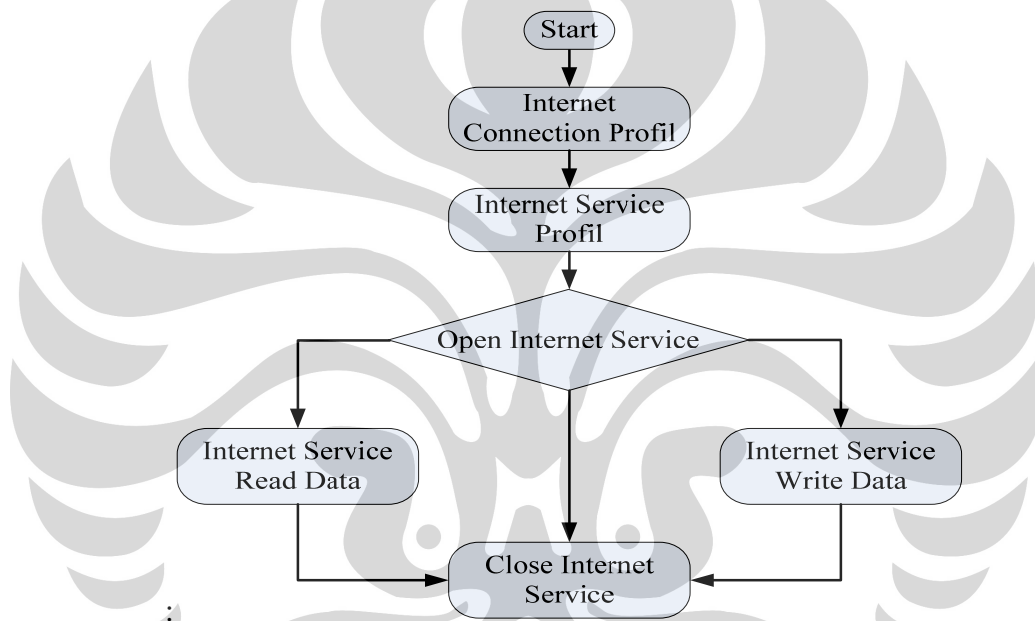




Gambar 3.2 Flowchart sistem secara keseluruhan

### 3.2 GPRS Gateway

Pada sistem ini GPRS gateway yang sifatnya mobile, tidak semua GSM Engine memiliki fitur internet service connection, dalam hal ini penulis menggunakan GSM Engine tipe MC55 buatan siemens yang telah support internet service via GPRS, adapun bahasa pemograman yang digunakan adalah AT Command untuk internet service, berikut merupakan diagram dari internet service via GPRS :



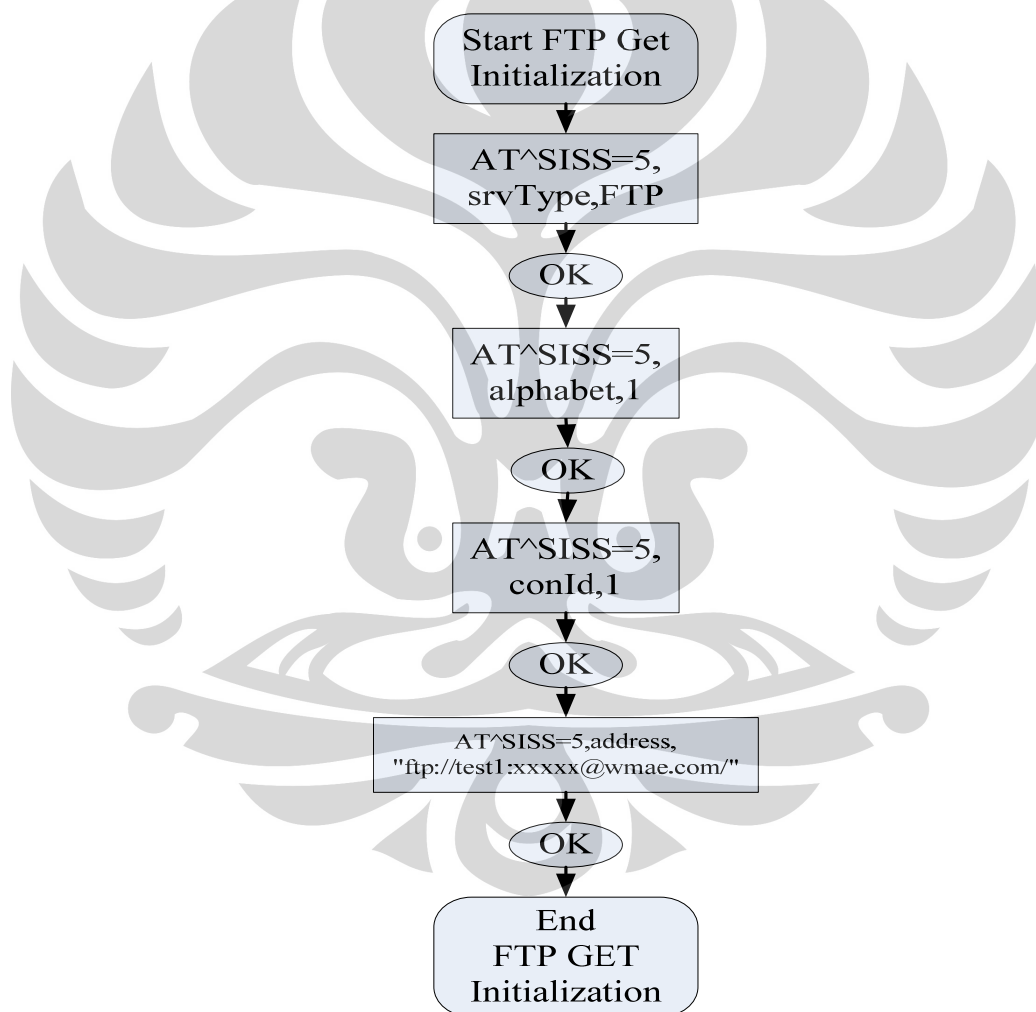
Gambar 3.3 Blok Diagram Internet Service Via GPRS

Blok diagram tersebut terdiri atas internet connection profil, internet service profil, open internet service dan close internet service. Internet connection profil merupakan pembuatan profil pada GSM Engine, dimana untuk masing – masing operator berbeda settingannya, untuk setting GPRS pada masing – masing operator dapat dilihat pada tabel 2.1.

### 3.2.1 Service Initialization

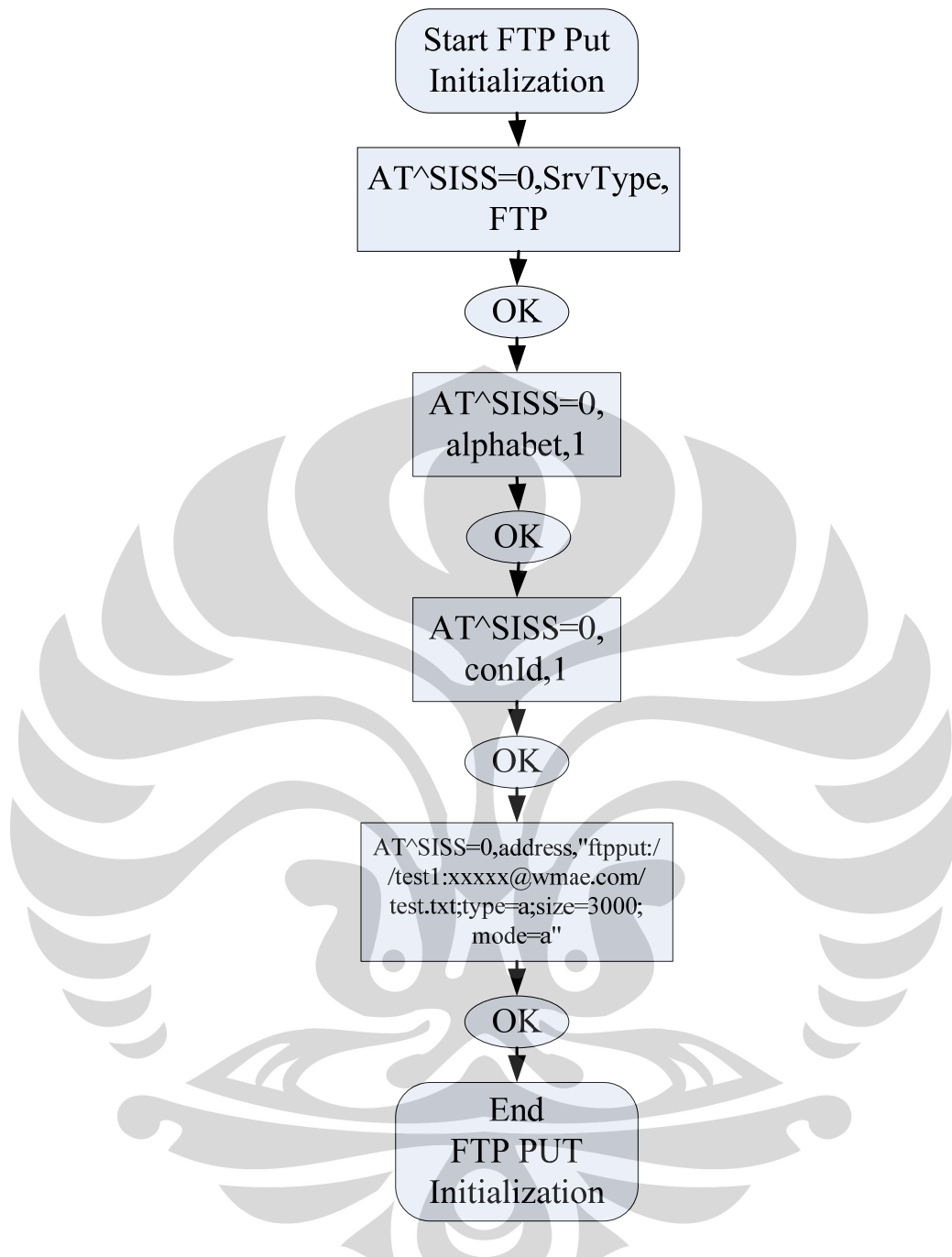
*Internet service profil* merupakan pembuatan profil service yang ingin digunakan, dalam hal ini terdiri dari socket connection, FTP, HTTP, SMTP dan POP3. open internet service merupakan koneksi GPRS ke internet, dalam hal ini kita dapat mengupload dan mendownload data. Sedangkan close internet service merupakan intruksi untuk keluar dari internet atau offline.

Berikut ini diagram untuk pembuatan profil FTP Get dan FTP Put.



Gambar 3.4 Diagram Pembuatan Profil FTP Get

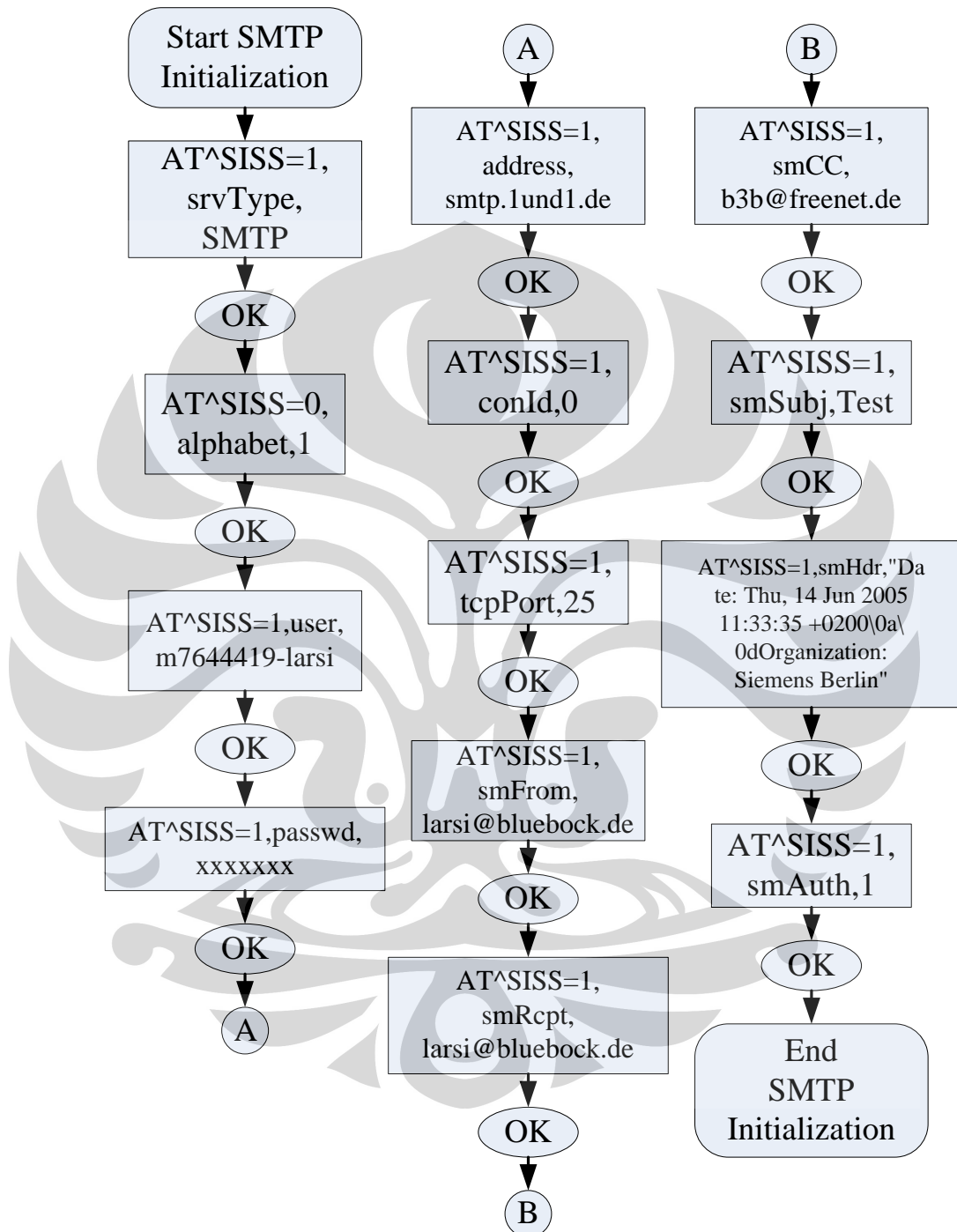




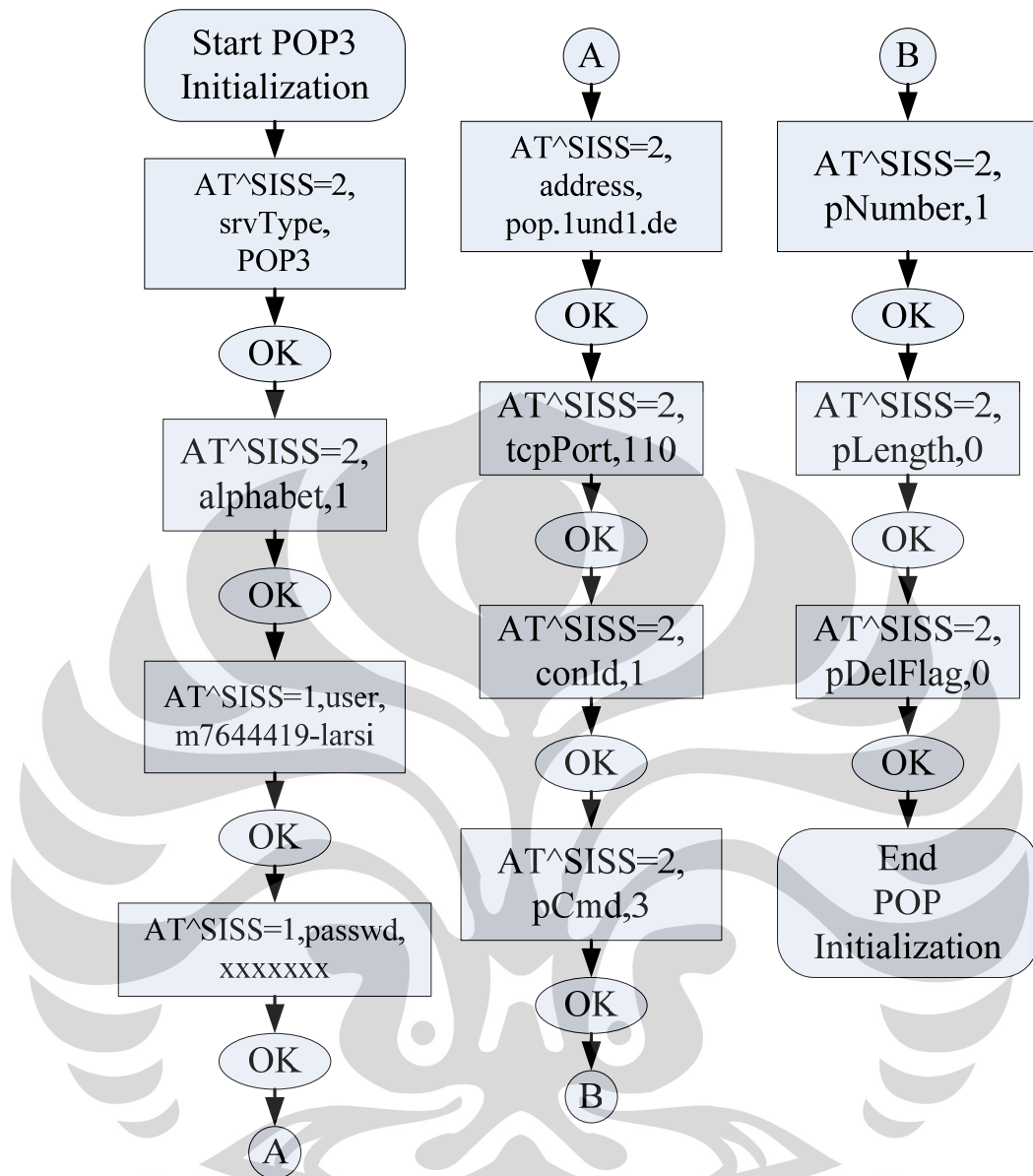
Gambar 3.5 Diagram Pembuatan Profil FTP Put

FTP (*File Transfer Protocol*) adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (file) komputer antar mesin-mesin dalam sebuah internetwork. FTP Get digunakan untuk mendownload file sedangkan FTP Put digunakan untuk

mengupload file ke server. Berikut ini adalah contoh pembuatan profil dari aplikasi email yaitu SMTP dan POP3.



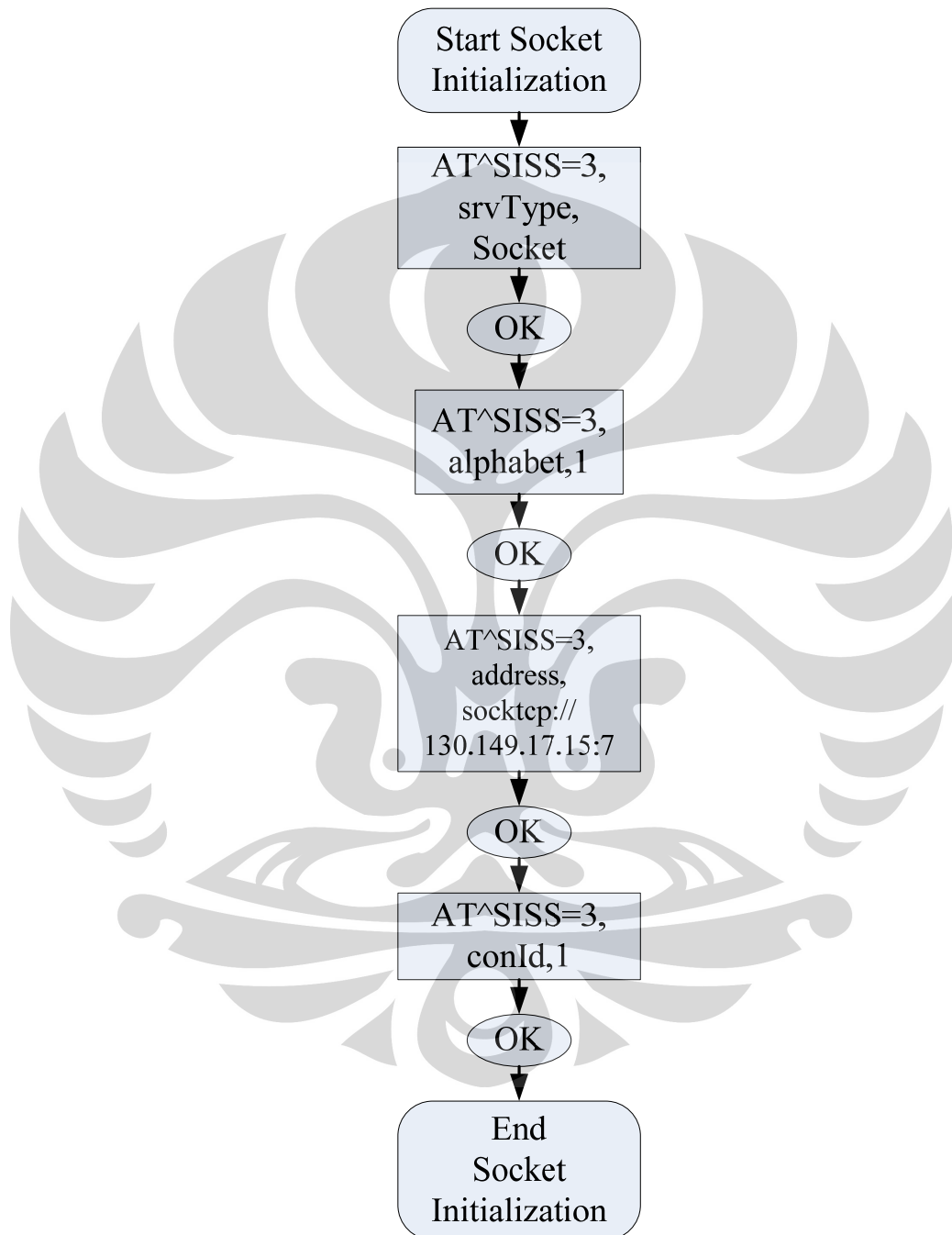
Gambar 3.6 Diagram Pembuatan Profil SMTP



Gambar 3.7 Diagram Pembuatan Profil POP3

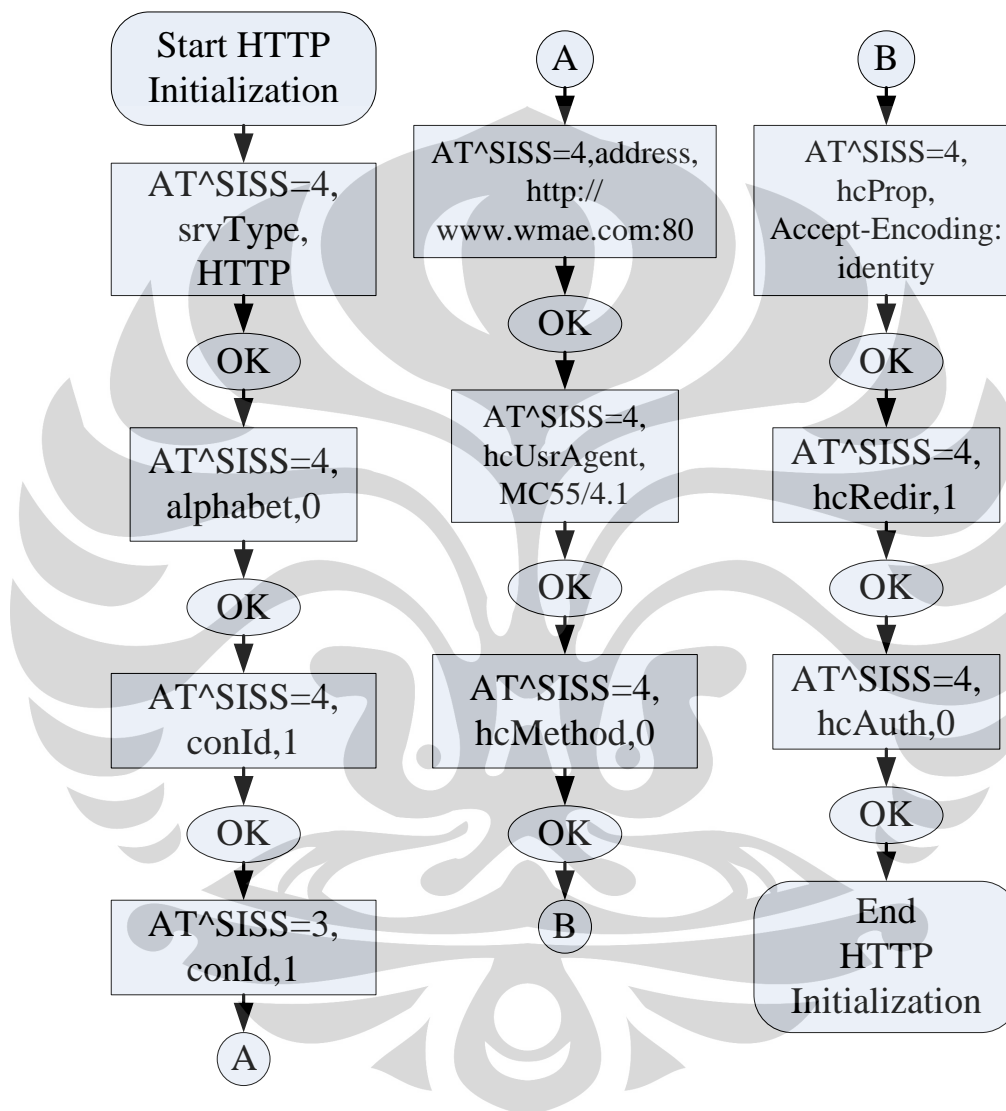
SMTP (*Simple Mail Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk pengiriman surat elektronik di Internet. Protokol ini dipergunakan untuk mengirimkan data dari komputer pengirim surat elektronik ke server surat elektronik penerima. Sedangkan POP3 (*Post Office Protocol version 3*) adalah protokol yang digunakan untuk mengambil surat elektronik (email) dari

server email. Untuk dapat menggunakan layanan ini web server harus sudah ada fasilitas POP3 dan SMTP, tidak semua hosting menyediakan fasilitas ini.



Gambar 3.8 Diagram Pembuatan Profil *Socket Connection*

UDP, singkatan dari *User Datagram Protocol*, adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara host-host dalam jaringan yang menggunakan TCP/IP.



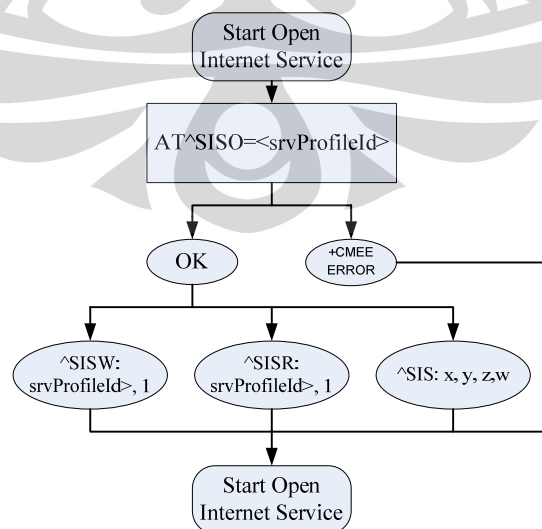
Gambar 3.9 Diagram Pembuatan Profil HTTP

HTTP (*HyperText Transfer Protocol*) adalah protokol yang dipergunakan untuk mentransfer dokumen dalam *World Wide Web* (WWW). Protokol ini adalah

protokol ringan, tidak berstatus dan generik yang dapat dipergunakan berbagai macam tipe dokumen.

### 3.2.2 *Open Internet Service*

*Open Internet Service* digunakan untuk mengaktifkan GSM Engine ke jaringan internet dengan menggunakan GPRS Network. Untuk dapat menggunakan ini kita harus membuat *profil internet service* terlebih dahulu, seperti pada gambar 3.3 sampai dengan gambar 3.8, perlu diketahui tidak semua profil dapat dibuat sekaligus di karenakan kemampuan GSM Engine yang terbatas, adapun maksimum profil yang mampu dibuat oleh GSM Engine adalah maksimum 10 sevice, dimana untuk HTTP maksimum 3 profil, Socket maksimum 6 profil, FTP maksimum 1 profil, SMTP maksimum 1 profil dan POP3 maksimum 1 profil. Apabila kita ingin menggunakan profil FTP Get untuk mendownload file dan FTP Put untuk mengupload file ke server dianjurkan untuk merestart ulang GSM Engine agar pada saat pembuatan profil berikutnya tidak terjadi error. Berikut merupakan blok diagram dari *Open Internet Service*.



Gambar 3.10 Diagram *Open Internet Service*

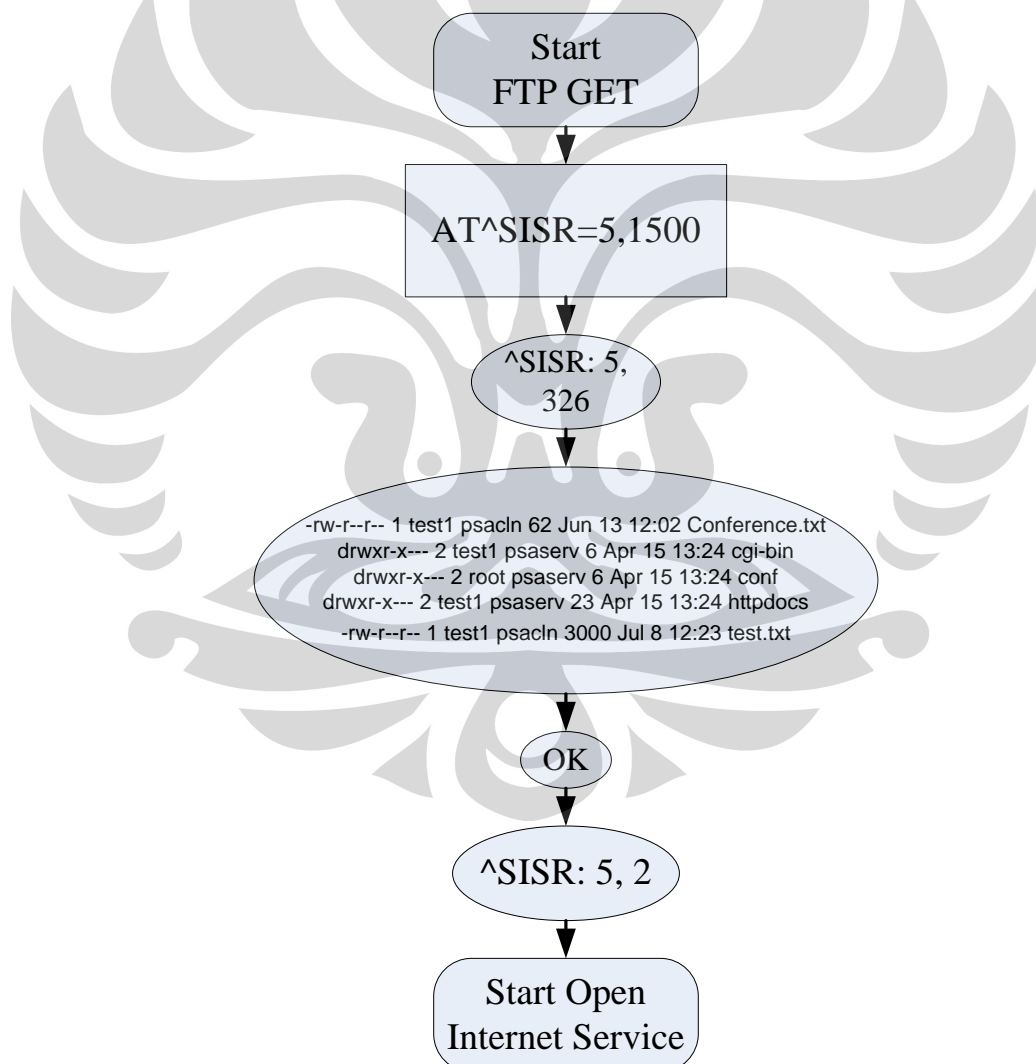
### 3.2.3 Read/Write Data

Read/Write Data merupakan prosedur yang digunakan untuk mengirim/membaca data dari internet. Adapun instruksi AT Command yang digunakan adalah :

`AT^SISR= <srvProfileId>,<reqReadLength>`

`AT^SISW= <srvProfileId>,<reqReadLength>`

AT^SISR digunakan untuk membaca data sedangkan AT^SISW digunakan untuk mengupload data ke server. Berikut merupakan contoh penggunaan *Read/Write* pada *FTP Connections*.



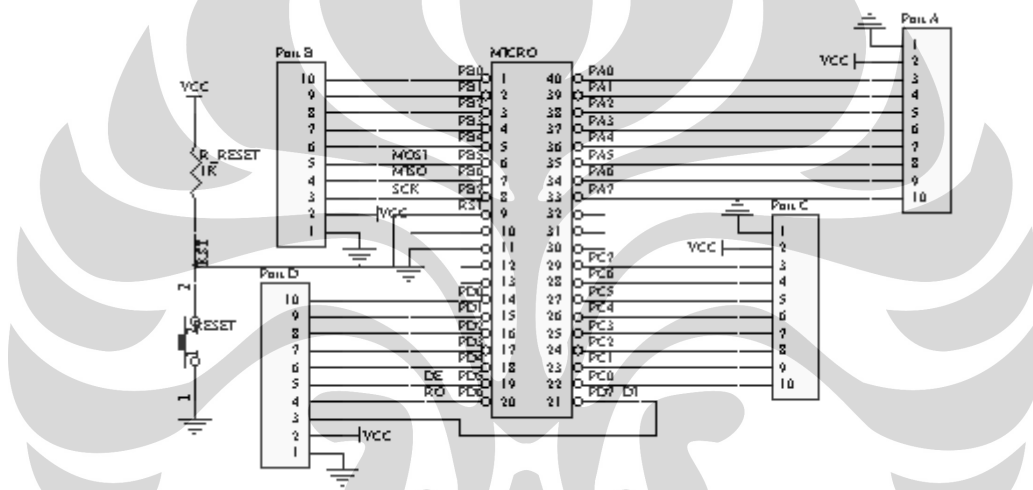
Gambar 3.11 Diagram *Read* Data Menggunakan FTP Get



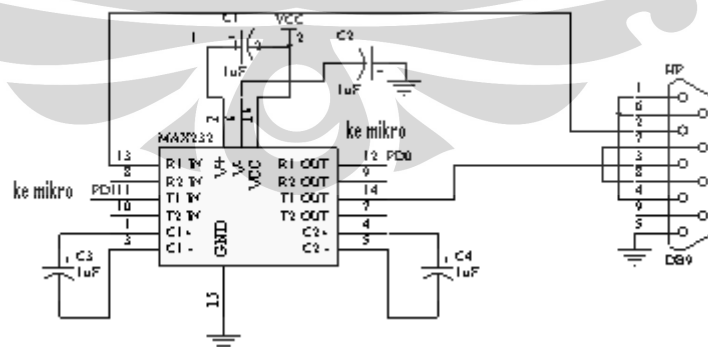


kompak dan pemrogramannya relatif lebih mudah. Pada gambar 3.13 menunjukkan rangkaian Minimum System (minsys) yang digunakan dalam rangkaian pengendali mikro (microcontroller).

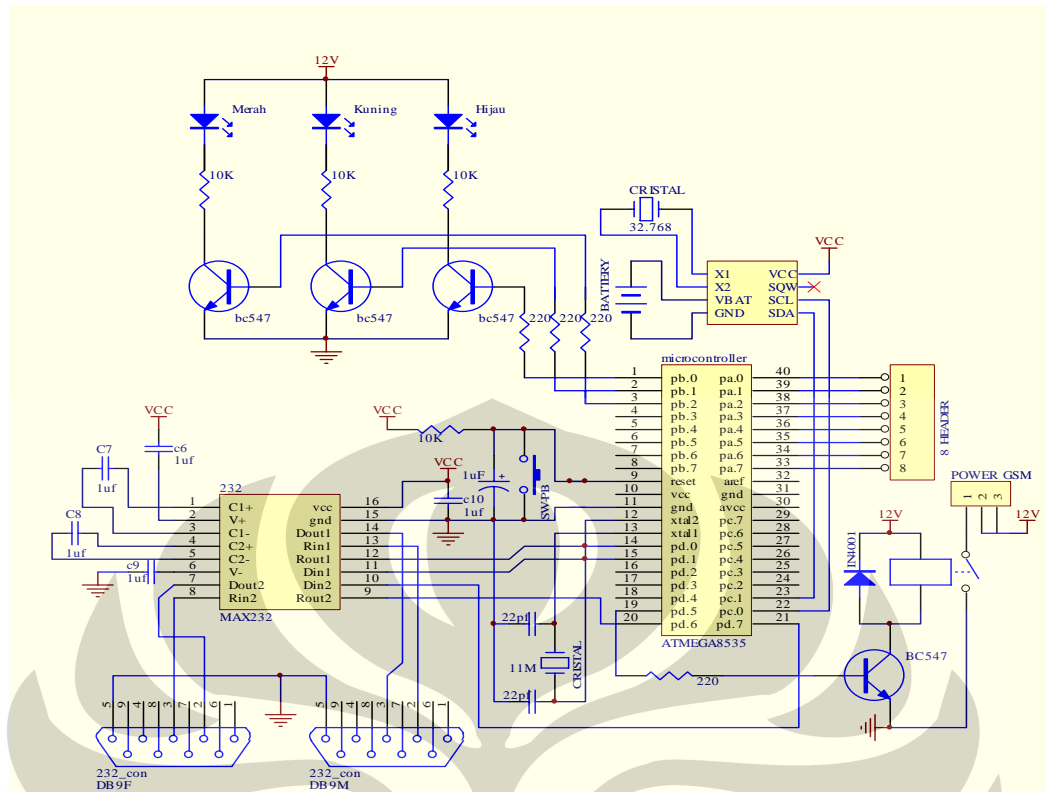
Pengendali mikro disini berfungsi untuk mengatur pembacaan, pengambilan dan pengiriman data ke server, Pengendali mikro disini bersifat dummy(diam), maksud dummy disini adalah pengendali mikro tidak melakukan instruksi apapun jika tidak ada permintaan/instruksi dari web server.



Gambar 3.13 Rangkaian minimum sistem

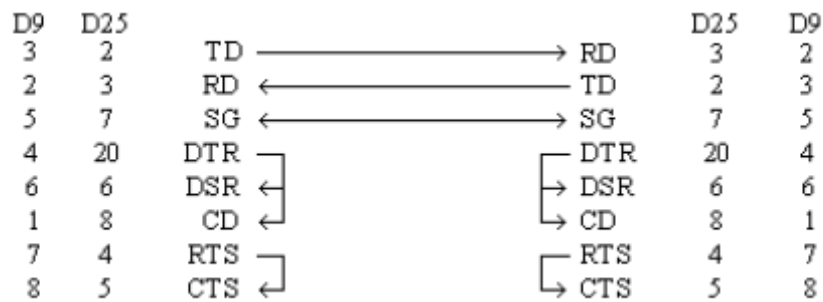


Gambar 3.14 Rangkaian komunikasi serial antara mikro dengan GSM Engine



Gambar 3.15 Rangkaian Secara Keseluruhan

Pada gambar diatas terdiri atas RTC, Rangkaian RS232, Rangkaian Aktuator, Rangkaian relay dan sebuah Port ADC. RTC yang digunakan adalah DS1307 komunikasi dengan mikro menggunakan komunikasi serial I2C, RTC disini berfungsi untuk memberitahukan waktu dan tanggal pada sistem hardware, Rangkaian komunikasi RS232 yang ada pada sistem terdiri dari 2COM (DB9F dan DB9M), DB9M digunakan sebagai penghubung antara mikro dengan GSM Engine MC55, sedangkan DB9F digunakan untuk dihubungkan ke perangkat lain, komunikasi serial RS232 dapat dilihat pada gambar berikut ini.

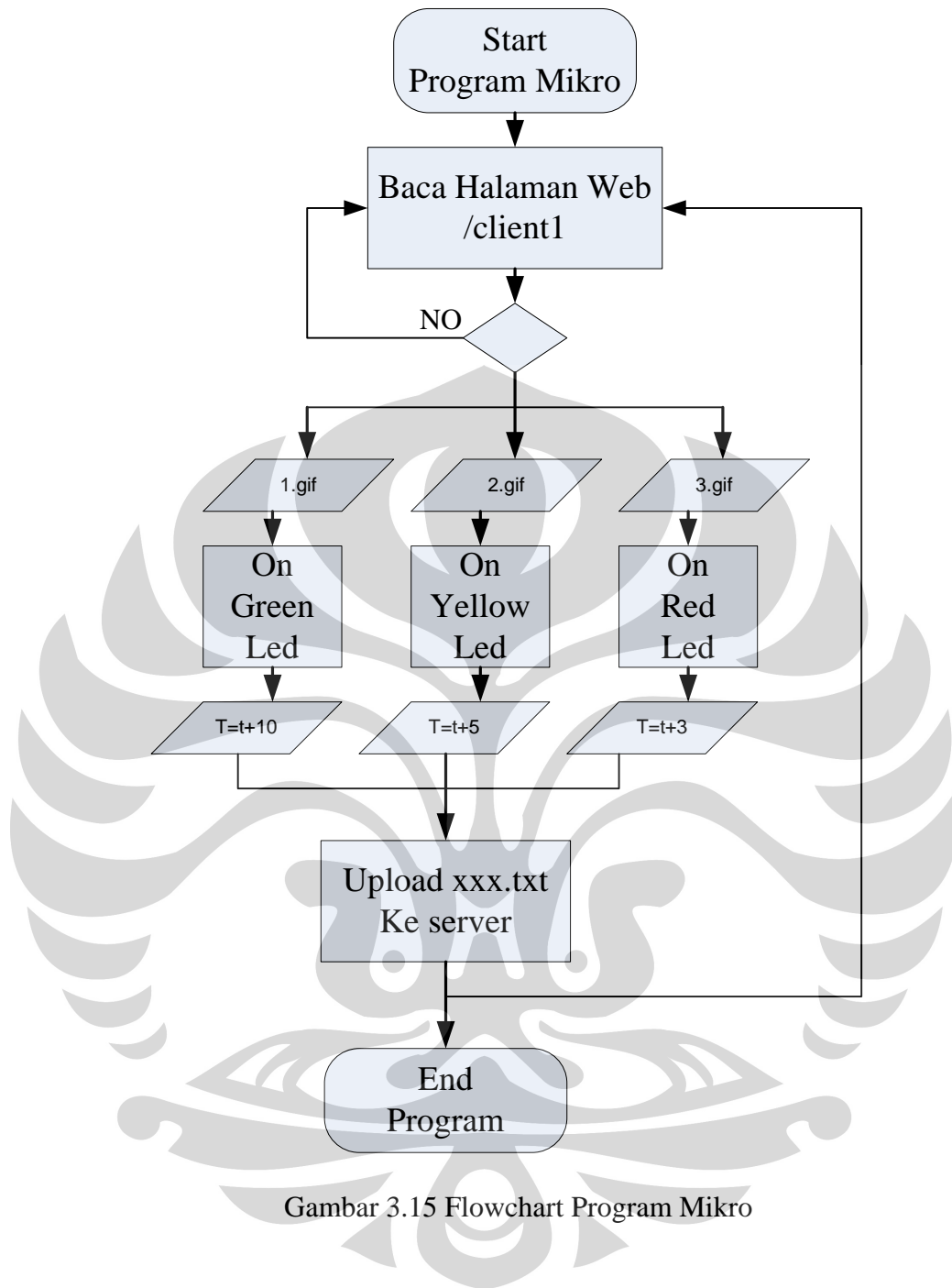


Gambar 3.16 Diagram Null Modem

Rangkaian aktuator yang terdiri dari 3 led berfungsi untuk memberitahukan status pembayaran dari si pelanggan, sedangkan rangkaian relay berfungsi untuk mengendalikan power dari GSM Engine agar penggunaannya dapat dikontrol oleh mikro.

Pengendali mikro ini juga sebagai suatu device yang dapat menterjemahkan isi pada hal web kedalam bentuk teks, adapun isi hal web pada /client1 dapat berupa 1.gif, 2.gif, 3.gif. masing – masing gambar (.gif) penggunaannya dikontrol oleh server.

Apabila pada hal web /client1 terdapat salah satu gambar maka mikro dapat langsung mengeksekusinya.



Gambar 3.15 Flowchart Program Mikro

Pada flowchart diatas dapat dilihat bahwa mikro hanya dapat mengeksekusi program selanjutnya apabila pada hal web /client1 terdapat sebuah gambar xxx.gif, apabila pada hal web /client1 tidak terdapat file gambar xxx.gif maka dilakukan pengecek lagi, dengan kata lain mikro bersifat pasif. Apabila pada hal web /client1 terdapat sebuah file 1.gif maka yang akan dieksekusi adalah mikro

akan mengirimkan file xxx.txt setiap 10menit, 2.gif mikro akan mengirimkan file xxx.txt setiap 5menit, dan 3.gif mikro akan mengirimkan file xxx.txt setiap 3menit

Adapun isi dari file xxx.txt adalah no pelanggan, tanggal dan jam, data dalam satuan watt dan daya yang digunakan oleh pelanggan.

### **3.3.1 Protokol Pengiriman data**

Mikro hanya dapat mengirim data apabila data permintaan dari server, permintaan tersebut ditunjukkan dengan adanya file data.gif pada halaman web /client1.

Format data yang dikirim adalah :

No Pelanggan PLN ; Tanggal Jam ; Daya yang Dikonsumsi ; Daya Pada Rumah  
No pelanggan berisi 12 digit decimal, daya yang dikonsumsi sudah dalam satuan watt, sedangkan daya pada rumah merupakan daya yang terpasang. Penggunaan “;” pada format data dimaksudkan untuk mempermudah pemisahan data, sehingga langsung masuk pada table database server.

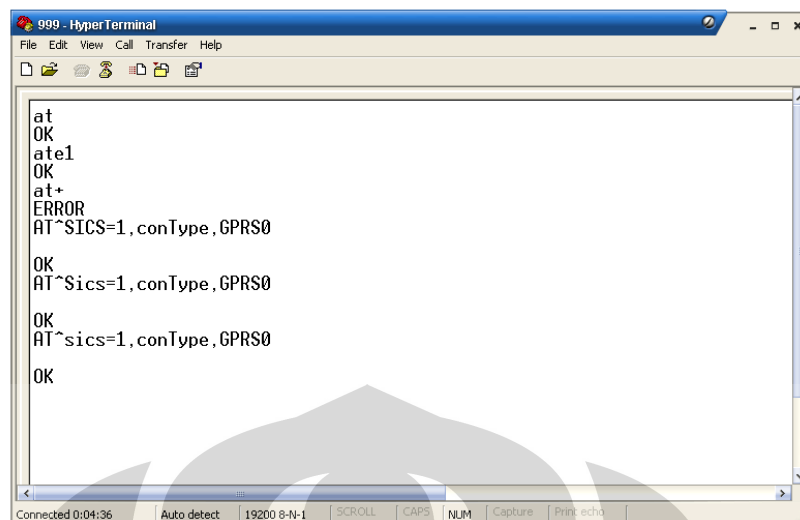
## **BAB 4**

### **DATA PENGUJIAN SISTEM**

Setelah keseluruhan sistem dibuat, maka langkah selanjutnya perlu dilakukan uji coba dan analisa sistem, apakah sistem dapat bekerja dengan baik dan benar. Pengujian dan pengambilan data ini dilakukan berdasarkan topik yang dibahas sehingga penulis hanya melakukan pengambilan data terhadap sistem yang penting.

#### **4.1 Pengujian *AT Command GSM Engine***

Pengujian ini bertujuan sebagai langkah awal dalam pembuatan sistem ini, karena data keberhasilan *AT command* pada GSM Engine sangat berpengaruh pada langkah berikutnya dalam pembuatan sistem ini. Jika fungsi dari *AT command* tidak berfungsi dengan baik maka GSM Engine tidak dapat digunakan untuk menjalankan sistem. Karena dalam fungsi *AT command* terdapat fungsi-fungsi yang dapat mengirim, membaca data dari halaman web. Pengujian fungsi *AT command* ini dilakukan dengan komunikasi serial antara GSM Engine dengan PC menggunakan program *Hyperterminal* yang sudah disediakan oleh *windows* pada PC. Dari hasil pengujian komunikasi serial antara GSM Engine dengan PC diperoleh data tampilan pada Gambar 4.1.



```

999 - HyperTerminal
File Edit View Call Transfer Help
at
OK
ate1
OK
at+
ERROR
AT^SICS=1,conType,GPRS0
OK
AT^Sics=1,conType,GPRS0
OK
AT^sics=1,conType,GPRS0
OK
Connected 0:04:36 | Auto detect | 19200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

```

Gambar 4.1 Data pengujian *AT command* GSM Engine pada Hyperterminal

Data diatas menunjukkan hasil pengujian *AT command* yang telah dilakukan. Dimana ada beberapa data pengujian yang ditunjukkan pada gambar yaitu yang pertama data pada baris pertama dan baris kedua yang menampilkan kata “AT dan ATE1” digunakan untuk mengetes komunikasi antara GSM Engine dengan PC, “OK dan ERROR” merupakan respon, “AT^SICS=1,conType,GPRS0” merupakan perintah untuk menyetting profil GSM, pada penulisan AT Command huruf “A” dan “T” harus sama, jika huruf “A” besar dan “t” ini dikirim dari mikro GSM Engine maka respon dari GSM Engine akan error.

#### 4.2 Pengujian AT Command Untuk Membuat *Internet Connection Setup Profil*

AT Command ini digunakan untuk membuat account GPRS pada GSM Engine, setting untuk setiap operator GSM berbeda, pada kesempatan ini penulis menggunakan operator dari Telkomsel, hasil pengujian pembuatan *Internet*

*Connection Setup Profil* GSM Engine dengan PC diperoleh data tampilan pada Gambar 4.2.



```
9999 - HyperTerminal
File Edit View Call Transfer Help
AT^SICS=1,conType,GPRS0
OK
AT^SICS=1,alphabet,1
OK
AT^SICS=1,user,wap
OK
AT^SICS=1,passwd,wap123
OK
AT^SICS=1,apn,www.wap.telkomsel.com
OK
OK
```

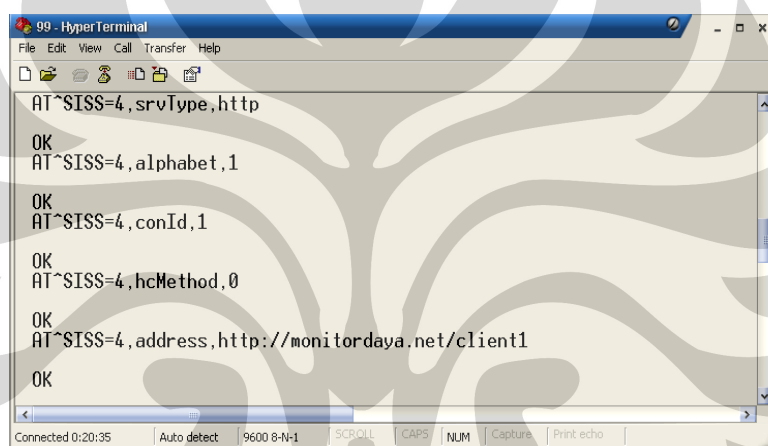
Gambar 4.2 Data pengujian *AT command Internet Connection Setup Profil*

Data diatas menunjukkan hasil pengujian AT command yang telah dilakukan. Dimana ada lima data pengujian, data tersebut dikirim dari PC ke GSM Engine. “AT^SICS=1,conType,GPRS0” digunakan untuk memerintahkan GSM Engine menggunakan koneksi GPRS, “AT^SICS=1,alphabet,1” digunakan untuk memerintahkan GSM Engine menggunakan karakter alphabet, “AT^SICS=1,user,wap” digunakan untuk memberi username pada profil GPRS dan instruksi “AT^SICS=1,passwd,wap123” digunakan untuk memberikan password pada profil GPRS, “AT^SICS=1,apn,www.wap.telkomsel.com” digunakan untuk memberikan *Access Point Name* pada profil GPRS



### 4.3 Pengujian AT Command Untuk Membuat *Internet Service Setup Profil*

AT Command ini digunakan untuk membuat *internet service* pada GSM Engine, pada kesempatan ini penulis menggunakan profil http, penulisan ini profil internet service ini harus didahului dengan penulisan *internet connection setup profil*, hal ini bertujuan ada pada saat dieksekusi atau di *run* (“AT^SISO=4”) tidak terjadi error, adapun instruksi pembuatan internet service setup profil dapat dilihat pada gambar 4.3



```

99 - HyperTerminal
File Edit View Call Transfer Help
AT^SISS=4,svrType,http
OK
AT^SISS=4,alphabet,1
OK
AT^SISS=4,conId,1
OK
AT^SISS=4,hcMethod,0
OK
AT^SISS=4,address,http://monitordaya.net/client1
OK
Connected 0:20:35 | Auto detect | 9600 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

```

Gambar 4.3 Data pengujian AT command *Internet Service Setup*

Data diatas menunjukkan hasil pengujian AT command yang telah dilakukan. Dimana ada empat data pengujian, terdiri atas

at^siss=4,svrType,Http

at^siss=4,conId,1

at^siss=4,hcMethod,0

at^siss=4,address,http://monitordaya.net/client1

Intruksi – instruksi tersebut digunakan untuk membuat profil http pada GSM Engine.

#### 4.4 Pengujian AT Command Untuk Mengeksekusi Profil

```

99 - HyperTerminal
File Edit View Call Transfer Help
OK
AT^SISO=4
OK
^SIS: 4, 0, 2201, "HTTP/1.1 301 Moved Permanently"
^SIS: 4, 0, 19, "Socket has been shutdown"
^SIS: 4, 0, 2200, "HTTP Redirect to:monitordaya.net:80/client1/"
^SIS: 4, 0, 2201, "HTTP/1.1 200 OK"
^SISR: 4, 1

```

Gambar 4.4 Data pengujian *AT Command* Untuk Mengeksekusi HTTP

Untuk mengeksekusi profil yang kita buat adalah “AT^SISO=<profil>”, ^SISR merupakan respon dari GSM Engine, dimana apabila terdapat respon ^SISR:0, 1 artinya halaman web telah siap untuk dibaca. Hasil pembacaan dapat dilihat pada gambar berikut ini:

```

99 - HyperTerminal
File Edit View Call Transfer Help
<h1>Index of /client1</h1>
<table>
  <tr>
    <th><a href="?C=N;0=D">Name</a></th>
    <th><a href="?C=M;0=A">Last modified</a></th>
    <th><a href="?C=S;0=F">Size</a></th>
    <th><a href="?C=D;0=A">Description</a></th></tr>
  <tr>
    <th colspan="5"><hr>
  </th>
  <tr>
    <td valign="top"></td>
    <td align="right"> - </td>
  </tr>
  <tr>
    <td valign="top"></td>
    <td><a href=?C=N;0=D">Name</a></td>
    <td><a href=?C=M;0=A">Last modified</a></td>
    <td><a href=?C=S;0=F">Size</a></td>
    <td><a href=?C=D;0=A">Description</a></td>
  </tr>
</table>
OK
AT^SISR=4,5
^SISR: 4, 5
1.gif

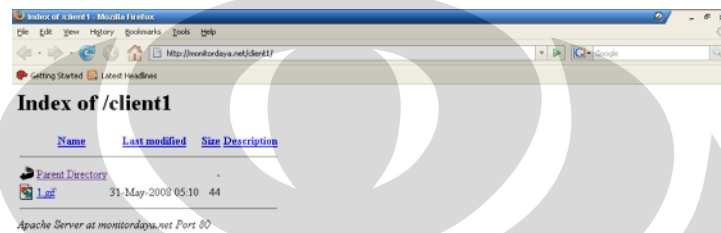
```

Gambar 4.5 Hasil Eksekusi Halaman Web

Dalam setiap pembacaan GPRS mempunyai kemampuan transfer 1,5 KB atas 1500 Karakter yang mampu dikirim setiap pakatnya, apabila kita ingin

melakukan pembacaan lagi maka instruksi “AT^SISR=0,1500” dapat dikirim kembali.

Berikut ini merupakan keyword untuk status pembayaran dari Pelanggan, dimana 1.gif berarti lunas, 2.gif berarti terlambat 1 bulan, 3.gif berarti terlambat 2 bulan, keyword ini dibaca dengan menggunakan protok HTTP, hasilnya sama pada gambar 4.4.



Gambar 4.6 Tanda Status Pembayaran Pelanggan



Gambar 4.7 Hasil file .txt yang dikirimkan melalui GSM Engine

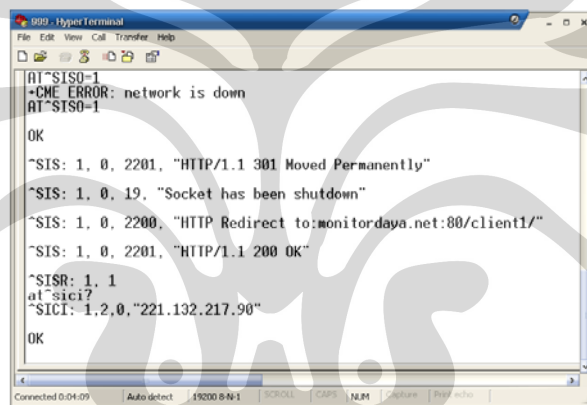
Metode pengiriman file .txt menggunakan protoko FTP, dimana saat pengiriman diberikan username dan password agar tidak ada orang yang menyalahgunakan, file .txt ini di generate oleh mikro, adapun isi dari file tersebut adalah:

No Pelanggan PLN ; Tanggal Jam ; Daya yang Dikonsumsi ; Daya Pada Rumah

#### 4.5 Pengujian AT Command Untuk Mengetahui IP Address

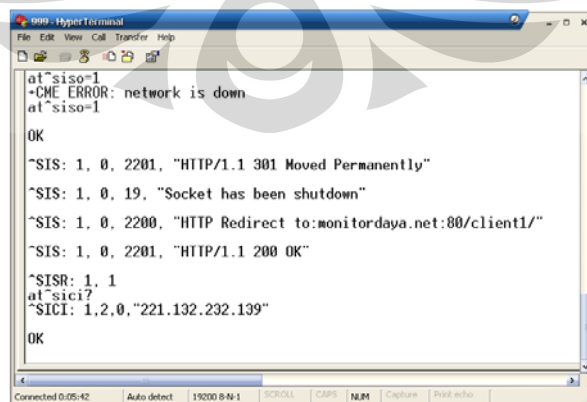
Pengujian dilakukan setelah GSM Engine terkoneksi ke salah satu profil internet service, adapun instruksi untuk mengetahui IP address nya adalah :

“AT^SICI?”



```
999 - HyperTerminal
File Edit View Call Transfer Help
-
~CME ERROR: network is down
AT^SISO-1
OK
^SIS: 1, 0, 2201, "HTTP/1.1 301 Moved Permanently"
^SIS: 1, 0, 19, "Socket has been shutdown"
^SIS: 1, 0, 2200, "HTTP Redirect to:monitordaya.net:80/client1/"
^SIS: 1, 0, 2201, "HTTP/1.1 200 OK"
^SISR: 1, 1
at^sici?
^SICI: 1,2,0,"221.132.217.90"
OK
Connected 0:04:09 Auto detect 19200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Gambar 4.8 Hasil Pengujian IP Address Pertama



```
899 - HyperTerminal
File Edit View Call Transfer Help
-
at^siso-1
~CME ERROR: network is down
at^siso-1
OK
^SIS: 1, 0, 2201, "HTTP/1.1 301 Moved Permanently"
^SIS: 1, 0, 19, "Socket has been shutdown"
^SIS: 1, 0, 2200, "HTTP Redirect to:monitordaya.net:80/client1/"
^SIS: 1, 0, 2201, "HTTP/1.1 200 OK"
^SISR: 1, 1
at^sici?
^SICI: 1,2,0,"221.132.232.139"
OK
Connected 0:05:42 Auto detect 19200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Gambar 4.9 Hasil Pengujian IP Address Kedua

Pada pengujian IP address pertama dan kedua terlihat bahwa IP address untuk masing – masing pengujian berbeda, untuk pengujian pertama IP Address pertama yaitu 221.132.217.90 dan yang kedua adalah 221.132.232.139. hasil ini menunjukkan bahwa IP Address yang diberikan oleh provider merupakan IP *Dynamic*.



## BAB 5

### KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang diperoleh penulis setelah melakukan penelitian tugas akhir serta saran-saran untuk perbaikan sistem dan hasil yang lebih baik lagi dimasa yang akan datang.

#### 5.1 Kesimpulan

Setelah menyelesaikan perancangan sistem serta melakukan pengujian terhadap sistem tersebut, maka penulis dapat menarik suatu kesimpulan bahwa:

- $AT^SICS=<conProfileId>, <conParmTag>, <conParmValue>$  digunakan untuk membuat *internet connections profile*.
- Pada *internet connections profile* kita dapat membuat profil CSD dan GPRS secara bersamaan.
- $AT^SISS=<servProfileId>, <conParmTag>, <conParmValue>$  digunakan untuk membuat *Internet Service Setup Profile*
- *Internet Service Setup Profile* terdiri atas FTP, HTTP, SMTP, POP3, dan UDP
- Koneksi GPRS tergantung pada jaringan GSM dan coverege, serta tergantung dari padatnya trafik data.
- $AT^SISO=<servProfileId>$  digunakan untuk mengaktifkan profil yang telah dibuat.
- $AT^SISR=<srvProfileId>, <reqReadLength>$  digunakan untuk membaca data dari

internet.

- $AT^{\wedge}SISR=<srvProfileId>$ ,  $<reqReadLength>$  digunakan untuk mengirim paket data ke internet.
- $AT^{\wedge}SISC=<servProfileId>$  digunakan untuk menutup *Internet Service*.
- Mikro secara aktif mengirim data secara periodik, dan setelah itu mikro mengecek status pembayarannya.
- IP Address pada GPRS bersifat *dynamic*
- Data *upload* dan *Download* untuk sekali transfernya maksimum 1500KB atau 1500 Karakter

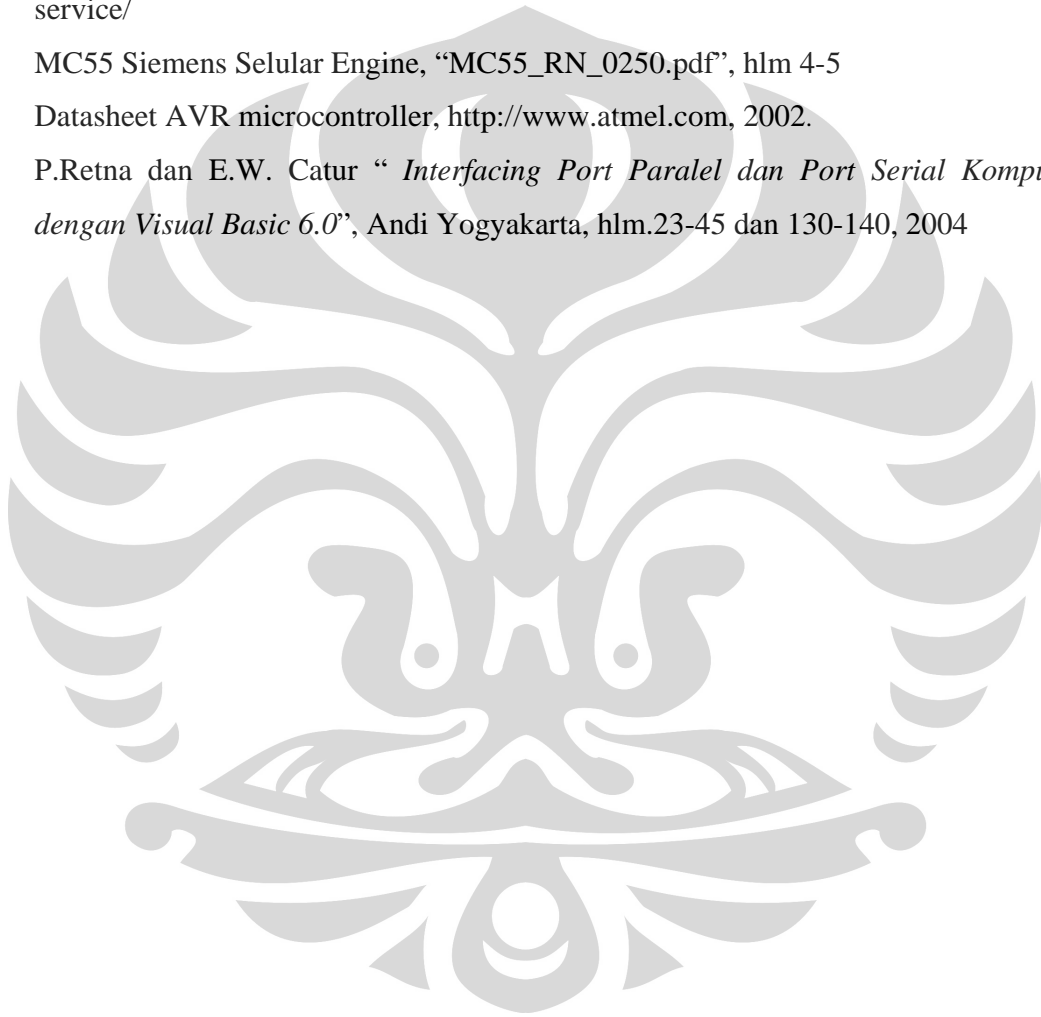
## 5.2 Saran

Berikut ini adalah saran dari penulis agar dimasa yang akan datang sistem ini dapat dibuat jauh lebih baik lagi:

- Sistem ini dibangun dengan menggunakan satu mikro dan 1 GSM Engine, diharapkan nantinya dapat digabung dengan beberapa mikro, untuk meminimalisir biaya yang dikeluarkan.
- Pada proses pengiriman file, nantinya file tersebut diharapkan langsung ke database bukan ke *directory* yang sekarang dilakukan.

## DAFTAR ACUAN

- [1] M55 GPRS Module, <http://www.gsmmodems.info/GSM%20Sitemap.htm>
- [2] GPRS, <http://id.wikipedia.org/wiki/gprs.htm>
- [3] Article GPRS, <http://mobileindonesia.net/2007/01/01/gprs-general-packet-radio-service/>
- [4] MC55 Siemens Selular Engine, “MC55\_RN\_0250.pdf”, hlm 4-5
- [5] Datasheet AVR microcontroller, <http://www.atmel.com>, 2002.
- [6] P.Retna dan E.W. Catur “ *Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0*”, Andi Yogyakarta, hlm.23-45 dan 130-140, 2004





# LAMPIRAN



## Information Elements Related to the Service Application

<urcInfoId>	<urcInfoText>	Description
<i>Information Elements Returned by the TCP/IP socket</i>		
2	Invalid socket descriptor	Socket error
3	Bad access specified	Socket error
4	Invalid operation	Socket error
5	No free socket descriptors	Socket error
6	The operation would block	Socket error
7	A previous attempt at this operation is still ongoing	Socket error
8	Socket is not bound	Socket error
9	The supplied buffer is too small / large	Socket error
10	Flags not supported	Socket error
11	Operation not supported	Socket error
12	The address is already in use	Socket error
13	The network is unavailable	Socket error
14	An established connection was aborted (transmission time-out or protocol error)	Socket error
15	Remote host has reset the connection	Socket error
16	No buffer space available	Socket error
17	The socket is already connected	Socket error
18	For TCP/IP sockets, the socket is not connected	Socket error
19	Socket has been shut down	Socket error
20	Connection timed out	Socket error
21	Remote host has rejected the connection	Socket error
22	Remote host is unreachable	Socket error
23	An unexpected error occurred	Socket error
24	Host not found	DNS error
25	An error occurred that may be transient; a further attempt may succeed.	DNS error
26	An unrecoverable error occurred	DNS error
<i>General Information Elements</i>		
46	Fatal: The service has detected an unknown error	
47	*)	Indicates that the remote service has closed the connection. The host shall close the service.

<code>&lt;urcInfoId&gt;</code>	<code>&lt;urcInfoText&gt;</code>	Description
48	Remote peer has closed the connection	Remote peer has closed the connection unexpectedly, and no data are available. The host shall close the service. Note that the " <code>^SIS</code> " URC only indicates the <code>&lt;urcInfoId&gt;</code> , but the <code>AT^SISE</code> command indicates both the <code>&lt;urcInfoId&gt;</code> and the <code>&lt;urcInfoText&gt;</code> .
49	Fatal: No memory is available for service action	
50	Fatal: Service has detected an internal error	
80		Connection profile not defined.
84	PPP LCP FAILED	PPP error
85	PAP Authentication failed	PPP error
86	CHAP Authentication failed	PPP error
87	IPCP failed	PPP error
<i>Warnings</i>		
4001		Applies only to interactive text mode: Indicates that number of input characters is greater than <code>&lt;cnfWriteLength&gt;</code> .
<i>General Information Elements</i>		
6001		General progress information for connection setup.

## Information Elements Related to FTP Service

<code>&lt;urcInfoId&gt;</code>	<code>&lt;urcInfoText&gt;</code>	Description
100	FTP Server rejects session.	
101	FTP Server rejects USER command.	
102	FTP Server rejects PASS command.	
103	FTP Server rejects PASV command or client can't detect remote IP address.	
104	FTP Server rejects requested client action.	
105	The given path segment is wrong.	
106	*)	The FTP client requests the file size from the server before starting the download. During transmission phase the client adds the bytes which are sent via serial line. After transmission is done the client compares the amount of bytes transferred via serial line and socket.
2100	FILE: <filename>	File name assigned by FTP server if mode=u is used.

## Information Elements Related to HTTP Service

<urcInfoId>	<urcInfoText>	Description
<i>Error URCS</i>		
200	HTTP-ERR: Not connected - cannot send request HTTP-ERR: failed on sending request	
201	<HTTP Response Str>	Client error
202	<HTTP Response Str> HTTP-ERR: Service unavailable	Server error
203	HTTP-ERR: Redirect failed - too many redirects HTTP-ERR: Redirect failed - not allowed HTTP-ERR: Redirect failed - location missing	Max. number of allowed redirects: 6
204	HTTP-ERR: auth failed - user name or password missing HTTP-ERR: auth failed - user name or password wrong HTTP-ERR: Authority required for this URL HTTP-ERR: No access to this URL allowed	
<i>Info URCS</i>		
2200	HTTP Redirect to: <Host>:<Port> <Path>	
2201	HTTP Response <Response code> Example: ^SIS: 2,3,2201, "HTTP Response: HTTP/1.1 200 OK"	

## Information Elements Related to POP3 Service

<urcInfoId>	<urcInfoText>	Description
<i>Errors</i>		
300	*)	Indicates that the POP3 User Command could not be executed successfully.
<i>Warnings</i>		
4300	*)	POP3 specific warning

## Information Elements Related to SMTP Service

<urcInfoId>	<urcInfoText>	Description
<i>Errors</i>		
400	*)	Indicates that the SMTP service could not be executed successfully.
<i>Warnings</i>		
4400	*)	SMTP specific warning

**Table 10.1:** Applicability of AT<sup>SICS</sup> <conParmTag> values

<conParmTag> value	CSD	GPRS0
"conType"	mandatory	mandatory
"user"	optional	optional
"passwd"	optional	optional
"apn"	ø	mandatory
"inactTO"	optional	optional
"calledNum"	mandatory	ø
"dataType"	mandatory	ø
"dns1"	optional	optional
"dns2"	optional	optional
"alphabet"	optional	optional

<conParmTag> <sup>(str)</sup>	
Internet connection parameter.	
"conType"	Type of Internet connection. For supported values of <conParmValue> refer to <conParmValue-conType>.
"alphabet"	Selects the character set for input and output of string parameters within a profile. The selected value is bound to the specific profile. This means that different profiles may use different alphabets. Unlike other parameters the alphabet can be changed no matter whether the <conParmTag> value "conType" has been set. For supported values of <conParmValue> refer to <conParmValue-alphabet>.
"user"	User name string: maximum 32 characters (where "" is default).
"passwd"	Password string: maximum 32 characters (where "" is default).
"apn"	Access point name string value: maximum 100 characters (where "" is default).
"inactTO"	Inactivity timeout value in seconds: 0 ... 2 <sup>16</sup> -1, default = 20 Number of seconds the bearer remains open although the service no longer needs the bearer connection.
"calledNum"	Called BCD number.
"dataType"	Data call type. For supported values of <conParmValue> refer to <conParmValue-dataType>.
"dns1"	Primary DNS server address (IP address in dotted-four-byte format). This value determines whether to use the DNS server addresses dynamically assigned by the network or a specific DNS server address given by the user. "dns1" = "0.0.0.0" (default) means that the CSD or GPRS connection profile uses dynamic DNS assignment. Any other address means that the Primary DNS is manually set. The default value applies automatically if no other address is set. Note that the AT <sup>SICS</sup> read command only returns a manually configured IP address, while the value "0.0.0.0" is not indicated at all, no matter whether assumed by default or explicitly specified. See also note below.

"dns2" Secondary DNS server address (IP address in dotted-four-byte format).  
If "dns1" = "0.0.0.0" this setting will be ignored. Otherwise this value can be used to manually configure an alternate server for the DNS1.  
If "dns1" is not equal "0.0.0.0" and no "dns2" address is given, then "dns2"="0.0.0.0" will be assumed automatically. The `AT^SICS` read command only returns a manually configured IP address, while the value "0.0.0.0" is not indicated at all, no matter whether assumed by default or explicitly specified.

`<conParmValue>`<sup>(str)</sup>

Parameter value; type and supported content depend on related `<conParmTag>`.

`<conParmValue-conType>`<sup>(str)</sup>

Supported connection type values in `<conParmValue>` for `<conParmTag>` value "conType".

"CSD" Circuit-switched data call.  
"GPRS0" GPRS connection.  
Settings of GPRS related commands are not used, e.g. `AT+CGDCONT`. When a service based on a GPRS connection profile is started after entering `AT^SISO` MC55 automatically tries to attach to the GPRS. Yet, the only exception is `AT+CGATT` which can be used any time to detach from the GPRS and, and thus disconnect the bearer opened with `AT^SISO`.  
"none" Clears the connection profile.

`<conParmValue-alphabet>`<sup>(str)</sup>

Character set selectable with `<conParmValue>` for `<conParmTag>` value "alphabet".

["0"] Character set determined with `AT+CSCS` applies.  
["1"] International Reference Alphabet (IRA, seven bit ASCII) applies.

`<conParmValue-dataType>`<sup>(str)</sup>

Supported data call type values in `<conParmValue>` for `<conParmTag>` value "dataType".

"0" ISDN  
["1"] Analog

**Table 10.2:** Applicability of AT<sup>^</sup>SISS <srvParmTag> values

<srvParmTag>	Mandatory or optional
<i>Socket service</i>	
"srvType"	mandatory
"conId"	mandatory
"alphabet"	optional
"address"	mandatory
"tcpMR"	optional
"tcpOT"	optional
<i>FTP service</i>	
"srvType"	mandatory
"conId"	mandatory
"alphabet"	optional
"address"	mandatory
"tcpMR"	optional
"tcpOT"	optional
<i>HTTP service</i>	
"srvType"	mandatory
"conId"	mandatory
"alphabet"	optional
"address"	mandatory
"user"	optional
"password"	optional
"hcContent"	optional
"hcContLen"	optional

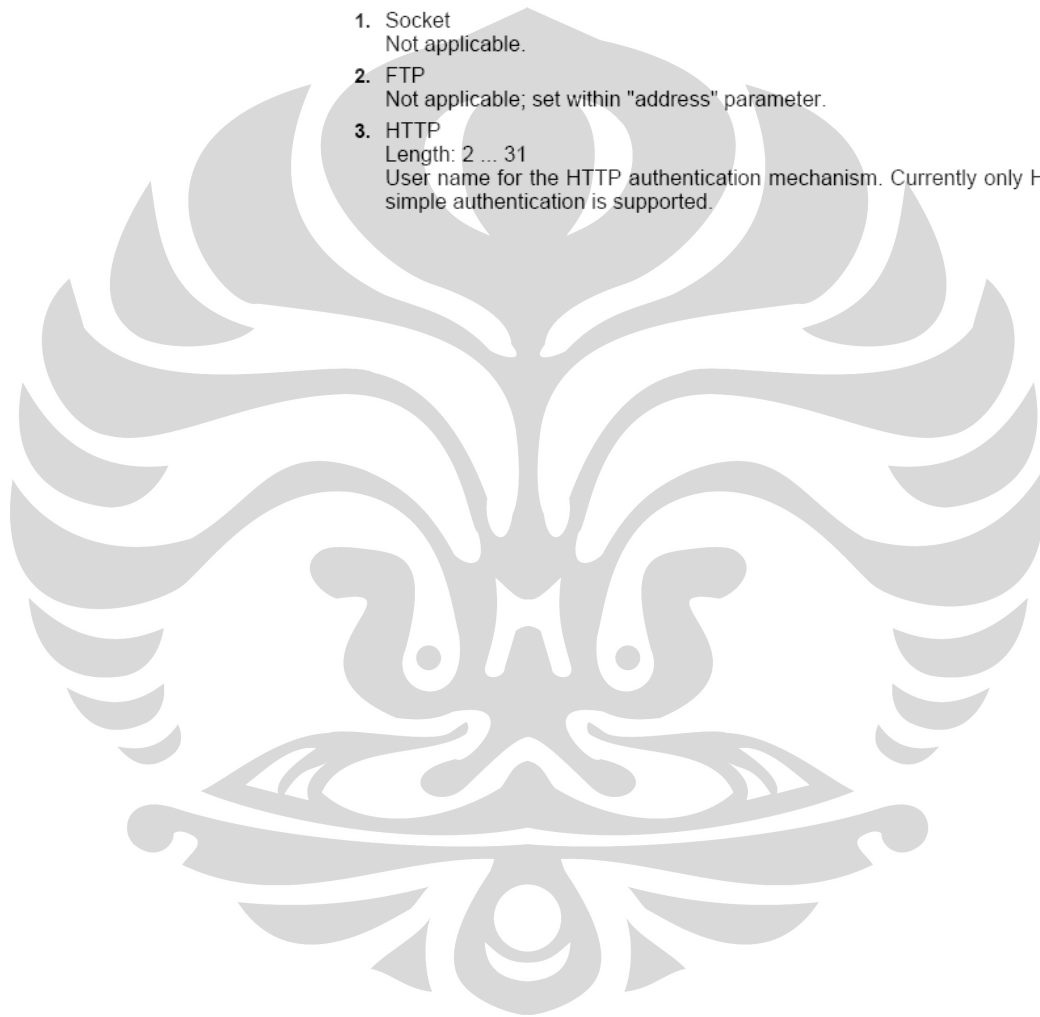
<b>&lt;srvParmTag&gt;</b>	Mandatory or optional
"hcUserAgent"	optional
"hcMethod"	mandatory
"hcProp"	optional
"hcRedir"	optional
"hcAuth"	optional
"tcpMR"	optional
"tcpOT"	optional
<i>SMTP service</i>	
"srvType"	mandatory
"conld"	mandatory
"alphabet"	optional
"address"	mandatory
"user"	optional
"password"	optional
"tcpPort"	optional
"smFrom"	mandatory
"smRcpt"	mandatory
"smCC"	optional
"smSubj"	optional
"smHdr"	optional
"smAuth"	optional
"tcpMR"	optional
"tcpOT"	optional
<i>POP3 service</i>	
"srvType"	mandatory
"conld"	mandatory
"alphabet"	optional
"address"	mandatory
"user"	mandatory
"password"	mandatory
"tcpPort"	optional
"pCmd"	mandatory
"pNumber"	optional
"pLength"	optional
"pDelFlag"	optional
"tcpMR"	optional
"tcpOT"	optional



## <srvParmTag>

Internet service profile parameter.

srvType	Type of Internet service to be configured with consecutive usage of <a href="#">AT^SISS</a> . For supported values of <srvParmValue> refer to <srvParmValue-srvType>.
alphabet	Selects the character set for input and output of string parameters within a profile. The selected value is bound to the specific profile. This means that different profiles may use different alphabets. Unlike other parameters the alphabet can be changed no matter whether the <srvParmTag> value "srvType" has been set. For supported values of <srvParmValue> refer to <srvParmValue-alphabet>.
user	User name string <ol style="list-style-type: none"><li>1. Socket Not applicable.</li><li>2. FTP Not applicable; set within "address" parameter.</li><li>3. HTTP Length: 2 ... 31 User name for the HTTP authentication mechanism. Currently only HTTP simple authentication is supported.</li></ol>



	<ol style="list-style-type: none"> <li>4. SMTP User name to be used for SMTP authentication (string). Length: 4 ... 64. If SMTP authentication is disabled, i.e. "smtpAuth" flag not set, user name parameter will be ignored.</li> <li>5. POP3 User name identifying a mailbox, i.e. mailbox name (string). Length: 1 ... 64.</li> </ol>
passwd	<p>Password string</p> <ol style="list-style-type: none"> <li>1. Socket Not applicable.</li> <li>2. FTP Not applicable; set within "address" parameter.</li> <li>3. HTTP Length: 2 ... 31 Password for the HTTP authentication mechanism. Currently HTTP simple authentication is supported only.</li> <li>4. SMTP Password to be used for SMTP authentication (string). Length: 4 ... 64. If SMTP authentication is disabled, i.e. "smtpAuth" flag not set, password parameter will be ignored.</li> <li>5. POP3 Server/mailbox-specific password (string). Length: 1 ... 64. Used to perform authentication with a POP3 server.</li> </ol>
conId	Internet connection profile to be used, for details refer <a href="#">AT^SICS</a> .
tcpPort	<p>TCP Port Number</p> <ol style="list-style-type: none"> <li>1. Socket Not applicable; set within "address" parameter.</li> <li>2. FTP Not applicable; set within "address" parameter.</li> <li>3. HTTP Not applicable; set within "address" parameter. If parameter is omitted the service connects to HTTP default port 80.</li> <li>4. SMTP SMTP server TCP port number (numeric) Length: 0 ... 2<sup>16</sup>-1 If this parameter is not set, SMTP default port number 25 is used.</li> <li>5. POP3 POP3 server TCP port number (numeric) Length: 0 ... 2<sup>16</sup>-1 If this parameter is not set, POP3 default port number 110 is used.</li> </ol>
address	<p>String value, depending on the service type either a URL in the case of Socket, FTP and HTTP or an address in the case of SMTP and POP3:</p> <ol style="list-style-type: none"> <li>1. Socket <ul style="list-style-type: none"> <li>- Socket type TCP client URL "socktcp://host':remote tcpPort'</li> <li>- Socket type TCP server URL "socktcp://listener:'local tcpPort' "</li> <li>- Socket type UDP client URL "sockudp://host':remote udpPort[:size='byte'][:port='word']" Parameter "size" (optional): 0: PDU size is variable (default). 1 ... 1460: Fixed PDU size in bytes.</li> </ul> </li> </ol>

Parameter "port" (optional):  
0: Port number will be assigned from service (default).  
1 ...  $2^{16}-1$ : defines the local port number for the UDP client.

## 2. FTP

- FTP client URL (get)  
"ftp://user':password'@'host':tcpPort/'url-path' [:type='a|i|d'] "  
Refer "IETF-RFC 1738".
- FTP client URL (put)  
"ftpput://user':password'@'host':tcpPort/'url-path'/element name'  
[:type='a|i'][:mode='u|a|d'] "

Used parameters:

"host" is mandatory, all other parameters are optional.

If "password" is set then "user" must be set as well.

If "user" is omitted the string "anonymous" is selected for "user" and "password".

If "password" is omitted the password request is served by an empty string.

If "tcpPort" is omitted the service connects to the FTP default port 21.

If "url-path" contains only the IP address a directory listing is requested.

If "url-path" contains the IP address and has a slash '/' appended a detailed directory listing is requested.

"type": [a]scii | i)image | d)irectory]

"mode": [u)nique | a)ppend | d)elele]

"u)nique" selects the FTP Store Unique command to create a file name unique to the current directory. If the file name is assigned by the server then the "[AT^SIS](#)" URC will appear, indicating [<urcInfoId>](#) 2100 and the file name.

"d)elele" clears given 'element name'.

If "mode" is omitted "replace mode" is default setting.

## 3. HTTP

HTTP client URL

Length: 6 ... 255

"http://server'/path':tcpPort' "

"server": FQDN or IP-address

"path": path of file or directory

"tcpPort": If parameter is omitted the service connects to HTTP default port 80.

Refer to "IETF-RFC 2616".

## 4. SMTP SMTP server address (string).

Length: 4 ... 256.

## 5. POP3 POP3 server address (string).

Length: 4 ... 256.

hcContent

Optional parameter for HTTP method "Post".

Length: 0 ... 127

Can be used to transfer a small amount of data. The content of this string will only be sent if "hcContLen" = 0. The maximum length of "hcContent" is 127 bytes.

To transmit a larger amount of data "hcContLen" must be set to a non-zero value. In this case the "hcContent" string will be ignored, and data transmission from the client to the server is done with [AT^SISW](#).

hcContLen

Mandatory parameter for HTTP method "Post".

Length: 0 ...  $2^{31}-1$

The content length shall be set in the header of the HTTP "Post" request before the data part is transferred.

If "hcContLen" = 0 then the data given in the "hcContent" string will be posted.

If "hcContLen" > 0 then the [AT^SISW](#) command will be used to send data from the client to the server. In this case, "hcContLen" specifies the total amount of data to be sent. The data can be sent in one or several parts. For each part, the transmission is triggered by the URC "[AT^SISW: x, 1](#)", then the [AT^SISW](#) write command can be executed. After the exact number of bytes are transferred via

the serial interface, the HTTP client will go from service state "Up" to service state "Closing" (see parameters `<srvState>` and `<srvState>` for detail). Finally, the URC "`^SISW: x, 2`" indicates that all data have been transferred and the service can be closed with `AT^SISC`.

hcUsrAgent	The user agent string must be set by the application to identify the mobile. Usually operation system and software version info is set with this browser identifier. Length: 0 ... 63
hcMethod	HTTP method specification: 0=GET, 1=POST, 2=HEAD.
hcProp	Parameter for several HTTP settings. Length: 0 ... 127 The general format is 'key': <space> 'value' "\0d\0a". Multiple settings can be given separated by "\0d\0a" sequences within the string. Possible 'key' values are defined at HTTP/1.1 Standard RFC 2616.
hcRedir	This flag controls the redirection mechanism of the MC55 acting as HTTP client (numeric). If "hcRedir" = 0: No redirection. If "hcRedir" = 1: The client automatically sends a new HTTP request if the server answers with a redirect code (range 30x). Default is 1.
hcAuth	If set ("hcAuth" = 1) this flag determines that the HTTP client will automatically answer on authentication requests from the server with the current "passwd" and "user" parameter settings. If these parameters are not specified the MC55 will terminate the HTTP connection and send an indication to the TA. Default is 1.
smFrom	Email sender address, i.e. "MAIL FROM" address (string). Length: 6 ... 256 A valid address parameter consists of local part and domain name delimited by a '@' character, e.g. "john.smith@somedomain.de".
smRcpt	Recipient address of the email, i.e. "RCPT TO" address (string). Length: 6 ... 256 If multiple recipient addresses are to be supplied the comma character is used as delimiter to separate individual address values, e.g. "john.smith@somedomain.de,tom.meier@somedomain.de".
smCC	CC recipient address of the email (string). Length: 6 ... 256 If multiple CC recipient addresses are to be supplied the comma character is used as delimiter to separate individual address values, e.g. "john.smith@somedomain.de,tom.meier@somedomain.de".
smSubj	Subject content of the email (string). Length: 0 ... 256 If no subject is supplied the email will be sent with an empty subject.
smHdr	This parameter, if set, will be appended at the end of the email header section (string). Length: 0... 256 Hence, it serves as a generic header field parameter which allows the user to provide any email header field. It is the user's responsibility to provide correct header fields! String of max. 256 characters. Example for multipart MIME messages: "Content-Type: multipart/mixed".
smAuth	SMTP authentication control flag (numeric). If disabled ( <code>&lt;srvParmValue&gt;</code> "0" by default), MC55 performs action without SMTP authentication. If enabled ( <code>&lt;srvParmValue&gt;</code> "1") authentication procedure with the SMTP server will be performed by means of supported authentication methods, using

values of "user" and "passwd" parameters. If MC55 and SMTP server are not able to negotiate an authentication mechanism supported by both parties, the MC55 continues action without authentication. MC55 supports SMTP authentication.

pCmd	POP3 user command to be executed by the POP3 service (numeric). For supported values of <code>&lt;srvParmValue&gt;</code> refer to <code>&lt;srvParmValue-pCmd&gt;</code> .
pNumber	Optional message number argument used by the POP3 commands List ("2"), Retrieve ("3") and Delete ("4"). For POP3 commands see <code>&lt;srvParmTag&gt;</code> value "pCmd". Length: 0 ... $2^{32}-1$ If no specific value is set in the service profile, the value "0" is assumed by default, i.e. "pNumber" is disabled.
pLength	Maximum message length (string, optional) Length: 0 ... $2^{32}-1$ "pLength" can be used to specify the length of the message(s) to be retrieved from or deleted on the POP3 server. If no specific value is set in the service profile, the default value "0" is assumed, which means that there is no limit on the message size. A warning will be issued in the following cases: <ul style="list-style-type: none"><li>• If "pNumber" &gt; 0 and a specific message to be retrieved from / deleted on the server is longer than "pLength".</li><li>• If "pNumber" = 0 and all messages to be retrieved from / deleted on the server are longer than "pLength".</li></ul> No warning will be issued in the following cases: <ul style="list-style-type: none"><li>• If there are no messages at all on the server.</li><li>• If the server has message(s) below <i>and</i> above the "pLength" specified. In this case, the message(s) within the range of "pLength" can be successfully retrieved or deleted, but the message(s) above "pLength" remain on the server without any further notification given to the user.</li><li>• Therefore, after retrieving / deleting messages, it is recommended to check the message status on the server. This can be done by adding a further POP3 service profile using the POP3 user command List ("2").</li></ul>
pDelFlag	Flag to be used with the POP3 user command Retrieve ("3"). Specifies whether or not to delete retrieved emails on the server (optional). For supported values of <code>&lt;srvParmValue&gt;</code> refer to <code>&lt;srvParmValue-pDelFlag&gt;</code> .
tcpMR	Parameter can be used to overwrite the global <code>AT^SCFG</code> parameter "Tcp/MaxRetransmissions" <code>&lt;tcpMr&gt;</code> for a specific Internet Service connection profile. If the parameter is not specified the value specified with <code>AT^SCFG</code> will be used. Supported values <code>&lt;srvParmValue&gt;</code> for this parameter are the same as described for <code>&lt;tcpMr&gt;</code> . Setting is not relevant for Internet Service "Socket" whit type "UDP".
tcpOT	Parameter can be used to overwrite the global <code>AT^SCFG</code> parameter "Tcp/OverallTimeout" <code>&lt;tcpOt&gt;</code> for a specific Internet Service connection profile. If the parameter is not specified the value specified with <code>AT^SCFG</code> will be used. Supported values <code>&lt;srvParmValue&gt;</code> for this parameter are the same as described for <code>&lt;tcpOt&gt;</code> . Setting is not relevant for Internet Service "Socket" whit type "UDP".

## AT^SISO Internet Service Open

- Upload (Socket, FTP, HTTP, SMTP):
  - Enter the **AT^SISO** command, e.g. AT^SISO=9.
  - Enter **AT^SISW**, specify **<reqWriteLength>**, e.g. AT^SISW=9,20. Check resulting response for **<cnfWriteLength>**. Optionally, check error with **AT^SISE**, e.g. AT^SISE=9. If necessary, repeat the sequence several times.
  - Enter last **AT^SISW** command and enable **<eodFlag>**, e.g. AT^SISW=9,0,1. <sup>1)</sup>  
If Socket service: Query available data with **AT^SISR**, e.g. AT^SISR=9,1430.
  - Check service state with **AT^SISI**, e.g. AT^SISI=9, if necessary several times until **<srvState>=6** ("Down").
  - Check error with **AT^SISE**, e.g. AT^SISE=9.
  - Close service with **AT^SISC**, e.g. AT^SISC=9.
- Download (Socket, FTP, HTTP, POP3):
  - Enter the **AT^SISO** command, e.g. AT^SISO=9.
  - Enter **AT^SISR**, specify **<reqReadLength>**, e.g. AT^SISR=9,1000. Check resulting response for **<cnfReadLength>**. If necessary, repeat the sequence several times until **<cnfReadLength>=-2** (end of data) or ERROR.
  - Check error with **AT^SISE**, e.g. AT^SISE=9.
  - Close service with **AT^SISC**, e.g. AT^SISC=9.
- Socket service (upload and download possible in one session):
  - Recommended: Set **AT+CMEE=2** to enable extended error text.
  - Enter the **AT^SISO** command, e.g. AT^SISO=9.
  - Enter **AT^SISR** or **AT^SISW**, specifying **<reqReadLength>** or **<reqWriteLength>**, e.g. AT^SISR=9,20 or AT^SISW=9,20. Check resulting response for **<cnfReadLength>** or **<cnfWriteLength>**. If necessary, repeat the sequence several times.
  - Check error with **AT^SISE**, e.g. AT^SISE=9.
  - If write action: Enter last **AT^SISW** command and enable **<eodFlag>**, e.g. AT^SISW=9,0,1. <sup>1)</sup>  
If Socket service: Query available data with **AT^SISR**, e.g. AT^SISR=9,1430.
  - Close service with **AT^SISC**, e.g. AT^SISC=9.

-----  
GPRS USING HTTP & FTP PROTOKOL  
CREATED BY : FARHAN NURDIN  
NPM : 0305220267  
-----

```
$regfile = "m8535.dat"  
$crystal = 8000000  
$baud = 9600  
$lib "mcsbyte.lbx"           ' for smaller code  
$lib "ds1307clock.lib"      ' modified lib  
$hwstack = 20  
$swstack = 8  
$framesize = 8  
  
Config Adc = Single , Prescaler = Auto , Reference = Internal  
Start Adc  
  
Config Lcd = 16 * 2  
Config Lcdpin = Pin , Db4 = Portc.5 , Db5 = Portc.4 , Db6 = Portc.3 , Db7 =  
Portc.2 , E = Portc.6 , Rs = Portc.7  
Config Portb = Output  
Reset Portb.0  
Reset Portb.1  
Reset Portb.2  
Config Portd.5 = Output  
Reset Portd.5  
Cursor Off  
  
'some subroutines  
Declare Sub Getline(s As String)  
Declare Sub Flushbuf()  
  
'Open Com Configurations  
Open "comd.7:9600,8,n,1" For Output As #2  
'Open "comd.6:9600,8,n,1" For Input As #1  
  
'used variables  
Dim I As Byte , B As Byte , C As String * 50  
Dim Sret As String * 200 , Stemp As String * 6  
Dim Pelanggan As String * 9  
Dim Daya As Single  
Dim Z As Byte  
Dim X As Byte  
Dim Y As Byte  
Dim D As Single  
Dim E As Single  
Dim F As Single  
Dim G As Byte
```

```

Dim H As Byte
Dim J As Integer
Dim K As Byte
Dim L As String * 1
Dim M As Word
Dim S As String * 1
Dim T As String * 1
S = String(1 , 59)
T = String(1 , 34)
Dim N As Single
Dim O As Byte
Dim R As Byte
Dim Gif As String * 5
Dim Gif1 As String * 5
Dim Gif2 As String * 5
Dim Hijau As Byte
Dim Rest As String * 50
Dim Status As String * 5
Dim Socket_error As String * 2
Dim Hija As Byte
Dim Adc_1 As Single
Dim W As Word

'we use a serial input buffer
Config Serialin = Buffered , Size = 12

'enable the interrupts because the serial input buffer works interrupts driven
Enable Interrupts

'define a constant to enable LCD feedback
Const Uselcd = 1
Config Portd.5 = Output
Upperline
Lcd " GPRS Gateway "
Waitms 500

Print "AT"
Print "AT"
Flushbuf
Print "ATE0"
'configure the scl and sda pins
Config Sda = Portc.1
Config Scl = Portc.0

'address of ds1307
Const Ds1307w = &HD0
Const Ds1307r = &HD1

```



```

Config Clock = User
'dim other needed variables
Dim Weekday As Byte

Print "DS1307"
Waitms 100
' assigning the time will call the SetTime routine
Time$ = "08:47:08"
Date$ = "06-09-08"

Pelanggan = "123456789"
'Socket_error = "^SIS: 1, 0, 18," + T + "Socket not connected" + T

N = 0
Cls
Upperline
Lcd " PROFIL GPRS "
Lowerline
Lcd " TELKOMSEL GPRS "
Wait 2
'Print #2 , Socket_error

-----
                          Profil GPRS USING TELKOMSEL
-----

Profil_gprs:
Print "AT^SICS=1,conType,GPRS0"
Print #2 , "AT^SICS=1,conType,GPRS0"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Profil_gprs
If Sret = "OK" Then Goto 1

1:
Print "AT^SICS=1,alphabet,1"
Print #2 , "AT^SICS=1,alphabet,1"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Profil_gprs
If Sret = "OK" Then Goto 2

2:
Print "AT^SICS=1,user,wap"
Print #2 , "AT^SICS=1,user,wap"
Wait 2

```

```
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Profil_gprs
If Sret = "OK" Then Goto 3
```

```
3:
Print "AT^SICS=1,passwd,wap123"
Print #2 , "AT^SICS=1,passwd,wap123"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Profil_gprs
If Sret = "OK" Then Goto 4
```

```
4:
Print "AT^SICS=1,apn,www.wap.telkomsel.com"
Print #2 , "AT^SICS=1,apn,www.wap.telkomsel.com"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Profil_gprs
If Sret = "OK" Then Goto 5
```

---

CLOSE SERVICE HTTP & FTP

---

```
5:
Flushbuf
Close_http_ftp:
Print "AT^SISC=1"
Print #2 , "AT^SISC=1"
Wait 2
Getline Sret
Print #2 , Sret
Print "AT^SISC=2"
Print #2 , "AT^SISC=2"
Wait 2
Getline Sret
Print #2 , Sret
```

---

CREATED HTTP PROFIL

---

```
Flushbuf
Cls
Upperline
```

```

Lcd " HTTP PROFIL "
Lowerline
Lcd " monitordaya.net"
Wait 1
Http_profil:
  Print "AT^SISS=1,svrType,http"
  Print #2 , "AT^SISS=1,svrType,http"
  Wait 2
  Getline Sret
  Print #2 , Sret
  If Sret = "ERROR" Then Goto Http_profil
  If Sret = "OK" Then Goto 6

6:
  Print "AT^SISS=1,conId,1"
  Print #2 , "AT^SISS=1,conId,1"
  Wait 2
  Getline Sret
  Print #2 , Sret
  If Sret = "ERROR" Then Goto Http_profil
  If Sret = "OK" Then Goto 7

7:
  Print "AT^SISS=1,hcRedir,1"
  Print #2 , "AT^SISS=1,hcRedir,1"
  Wait 2
  Getline Sret
  Print #2 , Sret
  If Sret = "ERROR" Then Goto Http_profil
  If Sret = "OK" Then Goto 8

8:
  Print "AT^SISS=1,hcMethod,0"
  Print #2 , "AT^SISS=1,hcMethod,0"
  Wait 2
  Getline Sret
  Print #2 , Sret
  If Sret = "ERROR" Then Goto Http_profil
  If Sret = "OK" Then Goto 9

9:
  Print "AT^SISS=1,address,http://monitordaya.net/client1"
  Print #2 , "AT^SISS=1,address,http://monitordaya.net/client1"
  Wait 2
  Getline Sret
  Print #2 , Sret
  If Sret = "ERROR" Then Goto Http_profil
  If Sret = "OK" Then Goto 10

```

---

CREATED FTP PROFIL

---

```
Flushbuf
10:
Cls
Upperline
Lcd " FTP PROFIL "
Lowerline
Lcd " monitordaya.net"
Wait 2
Ftp_profil:
Print "AT^SISS=2,svrType,ftp"
Print #2 , "AT^SISS=2,svrType,ftp"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Ftp_profil
If Sret = "OK" Then Goto 11

11:
Print "AT^SISS=2,alphabet,1"
Print #2 , "AT^SISS=2,alphabet,1"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Ftp_profil
If Sret = "OK" Then Goto 12

12:
Print "AT^SISS=2,conId,1"
Print #2 , "AT^SISS=2,conId,1"
Wait 2
Getline Sret
Print #2 , Sret
If Sret = "ERROR" Then Goto Ftp_profil
If Sret = "OK" Then Goto 13
```

---

OPEN HTTP PROFIL ([www.monitordaya.net/client1](http://www.monitordaya.net/client1))

---

```
Flushbuf
13:
Cls
Upperline
Lcd " OPEN /CLIENT1 "
Lowerline
```

Lcd " monitordaya.net"  
Wait 2

Print "AT^SISC=1"  
Print #2 , "AT^SISC=1"  
Wait 2  
Getline Sret  
Print #2 , Sret  
Print "AT^SISC=2"  
Print #2 , "AT^SISC=2"  
Wait 2  
Getline Sret  
Print #2 , Sret  
Flushbuf

Open\_http:  
Print "AT^SISO=1"  
Print #2 , "AT^SISO=1"  
Getline Sret  
Print #2 , Sret  
Socket\_error = Sret  
If Sret = "ERROR" Then Goto Close\_http\_ftp  
If Socket\_error = "^S" Then Goto Restart\_gsm  
Getline Sret  
Print #2 , Sret  
Getline Sret  
Print #2 , Sret  
Wait 2

---

READ HTTP PROFIL (www.monitordaya.net/client1)

---

Flushbuf  
14:  
Cls  
Upperline  
Lcd " READ /CLIENT1 "  
Lowerline  
Lcd " monitordaya.net"  
Wait 1  
Read\_client1:

Flushbuf  
Print "AT^SISR=1,644"  
Print #2 , "AT^SISR=1,644"  
Getline Sret  
Print #2 , Sret

If Sret = "ERROR" Then Goto 13

Getline Sret  
Print #2 , Sret  
Getline Sret  
Print #2 , Sret  
Getline Sret  
Print #2 , Sret  
Getline Sret  
Print #2 , Sret

Wait 2

Flushbuf

Flushbuf

Print "AT^SISR=1,5"

Print #2 , "AT^SISR=1,5"

Getline Sret

Print #2 , Sret

Getline Sret

Print #2 , Sret

Flushbuf

Gif = "1.gif"

Gif1 = "2.gif"

Gif2 = "3.gif"

If Gif = Sret Then Goto On\_hijau

If Gif1 = Sret Then Goto On\_kuning

If Gif2 = Sret Then Goto On\_merah

---

CLOSE SERVICE HTTP & FTP

---

Flushbuf

15:

Close\_http:

Print "AT^SISC=2"

Print #2 , "AT^SISC=2"

Getline Sret

Print #2 , Sret

Flushbuf

Print "AT^SISC=1"

Print #2 , "AT^SISC=1"

Getline Sret

Print #2 , Sret

Flushbuf

---

---

## MAIN OF PROGRAM

---

---

```
Main:
Do
W = Getadc(0)
  Adc_1 = W / 1023
  Adc_1 = Adc_1 * 450
  W = Adc_1
If Status = "1.gif" Then Gosub Cek1
If Status = "2.gif" Then Gosub Cek2
If Status = "3.gif" Then Gosub Cek3
```

```
Main1:
Cls
Upperline
Lcd " " ; Time$
Lowerline
Lcd W
Print #2 , Date$ ; " " ; Time$
Waitms 500
Loop
```

---

---

## CEK STATUS ([www.monitordaya.net/client1](http://www.monitordaya.net/client1))

---

---

```
Cek1:
If _min = 10 Then Gosub Hij10
If _min = 20 Then Gosub Hij20
If _min = 30 Then Gosub Hij30
If _min = 40 Then Gosub Hij40
If _min = 50 Then Gosub Hij50
If _min = 0 Then Gosub Hij60
Return

Hij10:
Print #2 , 10
Hija = 11
If Hija = _min Then Goto Ftp_setting
Return
```

```
Hij20:
Print #2 , 20
Hija = 21
If Hija = _min Then Goto Ftp_setting
Return
```

Hij30:  
Print #2 , 30  
Hija = 31  
If Hija = \_min Then Goto Ftp\_setting  
Return

Hij40:  
Print #2 , 40  
Hija = 41  
If Hija = \_min Then Goto Ftp\_setting  
Return

Hij50:  
Print #2 , 50  
Hija = 51  
If Hija = \_min Then Goto Ftp\_setting  
Return

Hij60:  
Print #2 , 60  
Hija = 1  
If Hija = \_min Then Goto Ftp\_setting  
Return

Cek2:  
If \_min = 5 Then Print #2 , "5"  
If \_min = 10 Then Print #2 , "10"  
If \_min = 15 Then Print #2 , "15"  
If \_min = 20 Then Print #2 , "20"  
If \_min = 25 Then Print #2 , "25"  
If \_min = 30 Then Print #2 , "30"  
If \_min = 35 Then Print #2 , "35"  
If \_min = 40 Then Print #2 , "40"  
If \_min = 45 Then Print #2 , "45"  
If \_min = 50 Then Print #2 , "50"  
If \_min = 55 Then Print #2 , "55"  
If \_min = 0 Then Print #2 , "60"  
Return

Cek3:  
If \_min = 3 Then Print #2 , "3"  
If \_min = 6 Then Print #2 , "6"  
If \_min = 9 Then Print #2 , "9"  
If \_min = 12 Then Print #2 , "12"  
If \_min = 15 Then Print #2 , "15"  
If \_min = 18 Then Print #2 , "18"  
If \_min = 21 Then Print #2 , "21"



```
If _min = 24 Then Print #2 , "24"  
If _min = 27 Then Print #2 , "27"  
If _min = 30 Then Print #2 , "30"  
If _min = 33 Then Print #2 , "33"  
If _min = 36 Then Print #2 , "36"  
If _min = 39 Then Print #2 , "39"  
If _min = 42 Then Print #2 , "42"  
If _min = 45 Then Print #2 , "45"  
If _min = 48 Then Print #2 , "48"  
If _min = 51 Then Print #2 , "51"  
If _min = 54 Then Print #2 , "54"  
If _min = 57 Then Print #2 , "57"  
If _min = 0 Then Print #2 , "60"  
Return
```

---

### RESTART GSM ENGINE

---

```
Restart_gsm:  
Cls  
Print #2 , "RESTART GSM"  
Upperline  
Lcd " RESTART MODEM "  
Lowerline  
Lcd "PLEASE WAIT....."  
Set Portd.5  
Wait 3  
Reset Portd.5  
Wait 6  
Goto 5
```

---

### CREATED FTP PROFIL ADDRESS

---

```
Ftp_setting:  
Print "AT^SISC=2"  
Print #2 , "AT^SISC=2"  
Wait 2  
Getline Sret  
Print #2 , Sret  
Print "AT^SISC=1"  
Print #2 , "AT^SISC=1"  
Wait 2  
Getline Sret  
Print #2 , Sret  
Flushbuf
```

```

Print "AT^SISS=2,address," ; T ;
"ftpput://data.monitordaya.net:monitor@data.monitordaya.net/www/" ; _day ;
_hour ; _min ; _sec ; ".txt" ; T
Print #2 , "AT^SISS=2,address," ; T ;
"ftpput://data.monitordaya.net:monitor@data.monitordaya.net/www/" ; _day ;
_hour ; _min ; _sec ; ".txt" ; T
Wait 2
Getline Sret
Print #2 , Sret
Print "AT^SISO=2"
Print #2 , "AT^SISO=2"
Getline Sret
Print #2 , Sret
Getline Sret
Print #2 , Sret
Print "AT^SISW=2,5"
Print #2 , "AT^SISW=2,5"

Getline Sret
Print #2 , Sret
Getline Sret
Print #2 , Sret
Wait 2
Print "AT^SISW=2,5"
Print #2 , "AT^SISW=2,5"

Getline Sret
Print #2 , Sret
Print W ; " "
Print #2 , Sret
Wait 2
Getline Sret
Print #2 , Sret

Flushbuf
Print "AT^SISW=2,0,1"
Print #2 , "AT^SISW=2,0,1"
Wait 2
Getline Sret
Print #2 , Sret
Getline Sret
Print #2 , Sret
Wait 2
Goto 13

```

STATUS (www.monitordaya.net/client1)

On\_hijau:

Status = "1.gif"

Reset Portb.1

Reset Portb.2

Set Portb.0

Cls

Upperline

Print #2 , "Lunas"

Lcd " Lunas "

Lowerline

Lcd " Terima Kasih "

Wait 1

Goto Main

On\_kuning:

Status = "2.gif"

Reset Portb.0

Reset Portb.2

Set Portb.1

Print #2 , " Nungguak 1 Bulan "

Cls

Upperline

Lcd "Nungguak 1 Bulan"

Lowerline

Lcd "Mohon di Lunasi"

Wait 1

Goto Main

On\_merah:

Status = "3.gif"

Reset Portb.0

Reset Portb.1

Set Portb.2

Print #2 , " Nungguak 2 Bulan "

Upperline

Lcd "Nungguak 2 Bulan"

Lowerline

Lcd "Mohon di Lunasi"

Wait 1

Goto Main

---

## RESPON GSM ENGINE

---

Sub Getline(s As String)

```
S = ""
Do
  B = Inkey()
  Select Case B
    Case 0
    Case 13
    Case 10 : If S <> "" Then Exit Do
    Case Else
      S = S + Chr(b)
  End Select
Loop
End Sub
```

'flush input buffer

```
Sub Flushbuf()
  Waitms 100
  Do
    B = Inkey()
  Loop Until B = 0
End Sub
```

---

## RTC DS1307

---

'called from ds1307clock.lib

Getdatetime:

```
I2cstart ' Generate start code
I2cwbyte Ds1307w ' send address
I2cwbyte 0 ' start address in 1307

I2cstart ' Generate start code
I2cwbyte Ds1307r ' send address
I2crbyte _sec , Ack
I2crbyte _min , Ack ' MINUTES
I2crbyte _hour , Ack ' Hours
I2crbyte Weekday , Ack ' Day of Week
I2crbyte _day , Ack ' Day of Month
I2crbyte _month , Ack ' Month of Year
I2crbyte _year , Nack ' Year
I2cstop
_sec = Makedec(_sec) : _min = Makedec(_min) : _hour = Makedec(_hour)
_day = Makedec(_day) : _month = Makedec(_month) : _year = Makedec(_year)
Return
```

Setdate:

```
_day = Makebcd(_day) : _month = Makebcd(_month) : _year = Makebcd(_year)
```

```
I2cstart ' Generate start code
```

```
I2cwbyte Ds1307w ' send address
```

```
I2cwbyte 4 ' starting address in 1307
```

```
I2cwbyte _day ' Send Data to SECONDS
```

```
I2cwbyte _month ' MINUTES
```

```
I2cwbyte _year ' Hours
```

```
I2cstop
```

Return

Settime:

```
_sec = Makebcd(_sec) : _min = Makebcd(_min) : _hour = Makebcd(_hour)
```

```
I2cstart ' Generate start code
```

```
I2cwbyte Ds1307w ' send address
```

```
I2cwbyte 0 ' starting address in 1307
```

```
I2cwbyte _sec ' Send Data to SECONDS
```

```
I2cwbyte _min ' MINUTES
```

```
I2cwbyte _hour ' Hours
```

```
I2cstop
```

Return

