



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Unand.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Unand.

**PERANCANGAN DAN PEMBUATAN PROTOTYPE SISTEM
PENDETEKSI KETINGGIAN AIR DAN KECEPATAN ANGIN LAUT
DENGAN LAYANAN SHORT MESSAGE SERVICE (SMS)**

SKRIPSI



**Benny Wilza Fiandra
No. BP : 06 975 006**

**FAKULTAS TEKNIK
UNIVERSITAS ANDALAS
PADANG 2008**

ABSTRAK

Untuk mendapatkan data level ketinggian air dan kecepatan angin laut dengan waktu lebih cepat dibutuhkan suatu alat yang dapat mengukur perubahan level ketinggian air dan kecepatan angin secara terus menerus. Alat tersebut dapat bermanfaat untuk peralatan peringatan dini terjadinya tsunami, data hasil pengukuran alat tersebut juga bermanfaat untuk nelayan dalam mengetahui ketinggian air dan kecepatan angin laut dapat lebih cepat dan akurat.

Penggunaan Short Message Service (SMS) dapat menyelesaikan masalah jarak dan tempat dalam pengiriman data hasil pengukuran. Data hasil pendeteksian level ketinggian air dan kecepatan angin diperoleh menggunakan rangkaian sensor ultrasonik dan sensor kecepatan dimana pusat pengendali adalah sebuah mikrokontroler AT89C52. Dalam proses pengiriman pesan SMS dari data hasil pengukuran ketinggian air dan kecepatan angin digunakan mode Protokol Data Unit (PDU).

Lamanya proses pengiriman pesan SMS level ketinggian air dan kecepatan angin dibutuhkan waktu rata-rata pada jam sibuk adalah sekitar 6.02 detik.

Kata kunci : *Short Message Service (SMS), sensor, dan mikrokontroler*

KATAPENGANTAR

Bismillaahirrahmaanirrahim,

Syukur alhamdulillah penulis ucapkan kehadiran Allah SWT. yang telah melimpahkan rahmat dan hidayah serta karunia-NYA kepada penulis sehingga setelah melewati beberapa hambatan dan rintangan, akhirnya penulis dapat menyelesaikan tugas akhir ini. Adapun judul Tugas Akhir adalah **“Perancangan dan Pembuatan Prototype Alat Pendeteksi Ketinggian Air dan Kecepatan Angin Laut Dengan Layanan SMS”**.

Adapun pembahasan dalam Tugas Akhir ini tidak luput dari kesalahan, kekeliruan dan kekurangan. Penulis sangat mengharapkan nasehat, bimbingan serta kritikan dari pembaca untuk kesempurnaan hasil pembahasan laporan ini. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan rasa terima kasih kepada :

1. Bapak Ikhwana Elfitri, MT selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Andalas Padang.
2. Bapak Ir. Darwison, MT dosen selaku pembimbing tugas akhir.
3. Dosen pengajar di Fakultas Teknik Elektro Jurusan Teknik Elektronika dan Telekomunikasi, atas ilmunya selama ini.
4. Sahabat dan rekan – rekan mahasiswa Teknik Elektronika Telekomunikasi baik secara langsung atau tidak langsung ikut membantu dalam menyelesaikan tugas akhir ini.
5. Rekan-rekan asisten Laboratorium Dasar Teknik Elektro.
6. Semua pihak yang telah membantu dalam penulisan laporan ini.

Akhir kata penulis berharap semoga tugas akhir ini memberikan manfaat bagi penulis khususnya dan pembaca umumnya.

Padang, Juli 2008

Penulis

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian.....	2
1.3 Manfaat Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Metodologi Penlisan	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN <i>SHORT MESSAGE SERVIS</i> (SMS) DAN PERANGKAT KERAS	6
2.1 <i>Short Message Servis</i> (SMS).....	6
2.1.1 Elemen-elemen SMS.....	7
2.1.2 Mekanisme <i>Store And Forward</i> pada SMS.....	9
2.1.2 Cara Kerja SMS.....	9
2.2 AT Command	10
2.3 Protokol Data Unit (PDU)	11
2.3.1 <i>Service Centre Address</i> (SCA).....	12
2.3.2 <i>Protokol Data Unit Type</i> (PDU Type)	13
2.3.2.1 <i>Reply Path</i> (RP).....	14
2.3.2.2 <i>User Data Header Identification</i> (UDHI).....	14
2.3.2.3 <i>Status Report Indicator</i> (SRI).....	14
2.3.2.4 <i>Status Report Request</i> (SRR)	15
2.3.2.5 <i>Validity Periode Format</i> (VPF)	15

2.3.2.6 <i>Reject Duplicate (RD)</i>	15
2.3.2.7 <i>More Message to Send (MMS)</i>	16
2.3.2.8 <i>Message Type Indicator (MTI)</i>	16
2.3.3 <i>Message Reference (MR)</i>	17
2.3.4 <i>Originator Address (OA) dan Destination Address (DA)</i>	17
2.2.5 <i>Protocol Identifier (PID)</i>	18
2.3.6 <i>Data Coding Scheme (DCS)</i>	18
2.3.7 Indikator waktu <i>Service Centre (SCTS)</i>	20
2.3.8 Masa Berlaku SMS (<i>Validity Period</i>)	20
2.3.9 <i>User Data Length (UDL) dan User Data (UD)</i>	21
2.3.10 Contoh Penggunaan PDU Pengiriman (SMS – <i>Submit</i>)	21
2.4 Pengkodean SMS	23
2.5 Sensor Ultrasonik	24
2.6 Sensor Kecepatan	25
2.7 <i>Liquid Crystal Display (LCD)</i>	26
2.8 Mikrokontroler AT89C52	27
2.8.1 Konsep Dasar Mikrokontroler	28
2.8.2 Bahasa Assembly MCS-51.....	28
2.8.3 Susunan Pin-pin Mikrokontroler AT89C52.....	29
BAB III PERANCANGAN SISTEM	31
3.1 Perancangan Perangkat Keras	32
3.1.1 Ponsel	32
3.1.2 Rangkaian Sensor	33
3.1.2.1 Rangkaian Sensor Ultrasonik.....	33
3.1.2.2 Rangkaian Sensor Kecepatan.....	34
3.1.3 Rangkaian Sistem Mikrokontroler	35
3.2 Perancangan Perangkat Lunak	36
3.3 Algoritma Perangkat Lunak	37
3.3.1 Sub Rutin Inisialisasi	37
3.3.2 Sub Rutin Sensor	38
3.3.3 Sub Rutin Pengiriman	40

BAB IV PENGUJIAN DAN HASIL	43
4.1 Pengujian Perangkat Keras	43
4.2 Pengujian Keseluruhan	47
4.2.1 Pengujian Pengiriman Pesan	48
4.5.2 Pengujian Level Ketinggian Air Turun	50
4.5.2 Pengujian Level Ketinggian Air Turun	51
BAB V KESIMPULAN DAN SARAN	53
5.1 Kesimpulan	53
5.2 Saran	53

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	Elemen pendukung SMS	7
Gambar 2.2	Struktur PDU Penerimaan (<i>SMS-DELIVER</i>)	12
Gambar 2.3	Struktur RDU Pengiriman (<i>SMS-SUBMIT</i>)	12
Gambar 2.4	Elemen-elemen penyusun <i>Service Centre Address</i>	13
Gambar 2.5	Tipe PDU untuk <i>SMS Submit</i>	13
Gambar 2.6	Tipe PDU untuk <i>SMS Deliver</i>	14
Gambar 2.7	Format PDU untuk nomor penerima dan nomor pengirim	17
Gambar 2.8	Konfigurasi bit <i>Data Coding Scheme</i> (DCS)	19
Gambar 2.9	Struktur PDU <i>User Data Length</i> (UDL) dan <i>User Data</i> (UD)	21
Gambar 2.10	Ilustrasi cara kerja sensor PING))) TM	25
Gambar 2.11	Simbol dioda infra merah dan photo transistor	26
Gambar 2.12	Susunan Pin (kaki) Mikrokontroler AT89C52	29
Gambar 3.1	Skema rancangan sistem	31
Gambar 3.2	Hubungan modul sensor ping dengan MC AT89C52	33
Gambar 3.3	Rangkaian sensor kecepatan	34
Gambar 3.4	Rangkaian sistem minimum mikrokontroler AT89C52	36
Gambar 3.5	Rangkaian Lengkap sistem	36
Gambar 3.6	Diagram alir perangkat lunak	37
Gambar 3.7	Algoritma sub sistem inialisasi	38
Gambar 3.8	Diagram alir sub rutin sensor	39
Gambar 3.9	Diagram alir sub rutin pengiriman	41

Gambar 3.10	Diagram alir pendeteksi ketinggian air dan kecepatan angin laut..	
	42
Gambar 4.1	Grafik pengukuran kecepatan angin.....	45
Gambar 4.2	Grafik perbandingan jarak benda terhadap waktu pantul.....	47

DAFTAR TABEL

Tabel 2.1	Beberapa perintah AT-Command	11
Tabel 2.2	Konfigurasi bit <i>Validity Periode Format</i>	15
Tabel 2.3	Konfigurasi bit MTI	17
Tabel 2.4	Konfigurasi bit ke-5 DCS	19
Tabel 2.5	Konfigurasi bit ke-4 DCS	19
Tabel 2.6	Konfigurasi bit ke-1 dan bit ke-0 DCS	20
Tabel 2.7	Konfigurasi bit DCS dan pengkodean yang digunakan	24
Tabel 2.8.	Konfigurasi pin- pin LCD.....	26
Tabel 4.1	Pengukuran parameter rangkaian sensor kecepatan.....	43
Tabel 4.2	Pengujian kecepatan angin dalam 5 sample pengukuran	44
Tabel 4.3	Perhitungan sensor ultrasonik	46
Tabel 4.4	Hasil pengujian pengiriman pesan dengan nomor pengirim berbeda	48
Tabel 4.5	Durasi pengiriman pesan dengan nomor pengirim yang sama.....	49
Tabel 4.6	Durasi pengiriman pesan dengan nomor pengirim berbeda.....	49
Tabel 4.7	Pengujian ketika level ketinggian air naik.....	50
Tabel 4.8	Pengujian ketika level ketinggian air turun.....	52

DAFTAR LAMPIRAN

Lampiran 1. Listing Program.....	56
Lampiran 2. Datasheet mikrokontroler AT89C52.....	74
Lampiran 3. Datasheet sensor Ping.....	81
Lampiran 4. Foto Alat.....	88

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya teknologi nirkabel (*wireless*), salah satunya adalah teknologi GSM (*Global System for Mobile Communications*), yang semakin murah dan dengan kapasitas jangkauan yang semakin luas, menyebabkan pemakaian telepon seluler tidak hanya berada pada salah satu golongan masyarakat tertentu saja, namun pemakai telepon seluler sudah menjangkau semua lapisan. SMS (*Short Message Service*) adalah salah satu fasilitas yang terdapat pada telepon seluler yang hampir setiap orang mengenalnya. Selain memiliki biaya operasional yang cukup murah, fasilitas ini juga merupakan media komunikasi dan sarana informasi antar individu yang cukup memiliki sifat waktu nyata (*real-time*).

Dengan terjadinya suatu bencana alam yang disebut tsunami yang mengakibatkan banyak korban jiwa meninggal, kejadian di akibatkan oleh adanya gelombang air laut yang besar. Banyaknya korban disebabkan masih kurangnya pengetahuan tentang kejadian tersebut dan belum adanya suatu alat ukur yang dapat mengetahui ketinggian air laut secara cepat dan tepat saat bencana terjadi. Dalam kehidupan nelayan sekarang, untuk mengetahui ketinggian air dan kecepatan angin laut masih berdasarkan perkiraan cuaca, dan juga suatu pekerjaan apabila memerlukan data pengukuran yang cepat dan akurat yang mana harus berada pada tempat kejadian, kadang kala ini merupakan hal yang dapat merugikan diri sendiri dan tidak efisien dalam segi tenaga, waktu dan biaya. Dari

permasalahan tersebut muncul ide untuk merancang suatu alat ukur jarak jauh yang lebih efektif dan efisien. Telepon seluler dengan fasilitas SMS yang mampu bertukar informasi berbasis teks secara jarak jauh (*remote*) dan tanpa kabel (*wireless*) dapat memberikan solusi yang tepat terhadap masalah alat ukur secara jarak jauh. Ditambah dengan dukungan teknologi mikrokontroler yang memungkinkan dibentuknya sebuah sistem komputer yang memiliki efisiensi daya dan tempat, menjadikan telepon seluler sebagai sarana alternatif selain sebagai sarana komunikasi juga dapat dijadikan sebagai sarana pengendali jarak jauh.

Berdasarkan ulasan diatas, penulis tertarik melakukan perancangan *prototype* tentang alat ukur ketinggian air dan kecepatan angin laut sebagai tugas akhir/skripsi. Untuk itu penulis mengajukan judul “ **Perancangan dan Pembuatan *Prototype* Alat Pendeteksi Ketinggian Air dan Kecepatan Angin Laut Dengan Layanan *Short Message Service* (SMS)**”.

1.2 Tujuan Penelitian

Adapun tujuan dari penelitian adalah sebagai berikut :

1. Merancang sebuah *prototype* sistem pengukuran level ketinggian air dan kecepatan angin yang bersifat *mobile*.
2. Melakukan pengujian perubahan level ketinggian air dalam bak air 30x20x12 cm³.

1.3 Manfaat Penelitian

Manfaat yang dapat diperoleh dari tugas akhir adalah sebagai berikut:

1. Menciptakan sebuah alat pengukuran yang bersifat *mobile* dengan biaya yang relatif murah.
2. Dapat digunakan untuk alat pendeteksian dini terjadinya stunami.
3. Mengetahui perubahan level ketinggian air dan kecepatan angin laut dengan waktu cepat bagi para nelayan.

1.4 Batasan Masalah

Agar pembuatan tugas akhir lebih terarah, maka dibuatlah batasan masalah sebagai berikut :

1. Mikrokontroler yang digunakan adalah keluarga MCS51 yaitu AT89C52.
2. Ponsel yang digunakan adalah merek Siemens C35/C45
3. Mode transfer data SMS adalah mode PDU.
4. Pengujian *prototype* dilakukan dalam bak air 30x20x12 cm dan menggunakan kipas angin kecil.
5. Operator telekomunikasi yang digunakan adalah Telkomsel

1.5 Metodologi Penulisan

Metode penelitian yang digunakan dalam perancangan ini adalah

1. Studi literatur tentang sistem komunikasi dengan menggunakan SMS dan pemrograman.

2. Perancangan rangkaian yang akan digunakan dalam sistim ini yaitu rangkaian sistim minimum mikrokontroler dan rangkaian sensor.
3. Pembuatan program pendeteksi ketinggian air dan kecepatan angin.
4. Pengujian sistem
5. Penulisan laporan tugas akhir.

1.6 Sistematika Penulisan

Agar lebih terstruktur laporan ini maka ditulis dengan sistematika sebagai berikut :

Bab I Pendahuluan

Pendahuluan berisikan tentang latar belakang, identifikasi masalah, batasan masalah, manfaat penelitian, tujuan penelitian, metoda penelitian, dan sistematika penulisan laporan

Bab II Tinjauan Pustaka

Tinjauan pustaka berisikan tentang SMS, konsep-konsep pengiriman dan penerimaan SMS, sensor ultrasonik, sensor optocoupler, LCD serta mikrokontroler AT89C52.

Bab III Perancangan Sistem

Perancangan dan pembuatan ini meliputi perancangan dan pembuatan sistem pendeteksi ketinggian air dan kecepatan angin berupa perangkat keras (*hardware*) yang terdiri dari beberapa bagian yaitu rangkaian komunikasi serial, rangkaian sensor, rangkaian pengendali, serta perancangan perangkat lunak

(*software*) untuk membaca perubahan level ketinggian air dan kecepatan angin serta pengiriman pesan.

Bab IV Pengujian dan Hasil

Pengujian dan hasil berisikan pengujian seluruh subsistem baik secara terpisah maupun secara terintegrasi serta berisi hasil dari pengujian yang telah dilakukan.

Bab V Penutup

Penutup berisikan kesimpulan dan saran untuk dapat dijadikan bahan pertimbangan dalam pengembangan perancangan di masa yang akan datang.

Lampiran Lampiran berisikan listing program, data sheet komponen yang digunakan dalam perancangan sistem pendeteksi ketinggian air kecepatan angin serta gambar rangkaian sistem ini secara keseluruhan.

BAB II

TINJAUAN *SHORT MESSAGE SERVICE* (SMS) DAN PERANGKAT KERAS

Penelitian sebelumnya dilakukan oleh adalah Husnil Kamil (2006) dengan judul “*Penggunaan Short Message Service (SMS) sebagai Media Transfer Perintah Operasi Motor DC*” hasil dari penelitian adalah waktu rata-rata yang dibutuhkan untuk mengirim SMS sampai SMS diterima oleh telepon seluler yang lainnya sekitar 13.334 detik dan waktu yang dibutuhkan untuk memproses SMS sekitar 17,704 detik. Dan Iroldi Febri (2007) dengan judul “*Perancangan dan Pembuatan Sistem Monitoring Suhu Menggunakan Short Message Service (SMS) Berbasis Mikrokontroler AVR ATmega8535*”. Hasil penelitian adalah waktu yang digunakan untuk pengontrolan suhu dalam proses pengiriman dan penerimaan SMS adalah sekitar 6 detik.

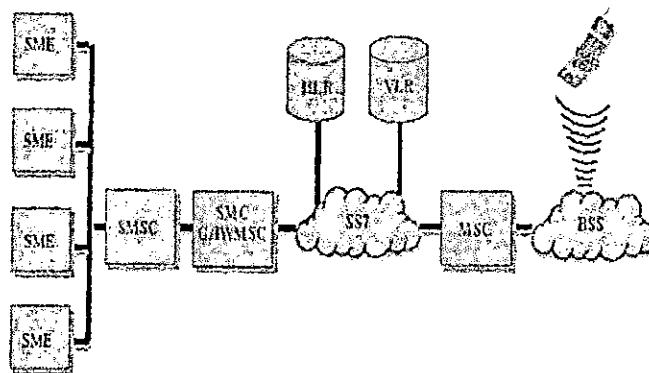
2.1 SMS (*Short Message Service*)

Short Message Service adalah salah satu layanan pesan berupa teks dari perusahaan operator telephone selular GSM, yang dikembangkan dan distandarisasi oleh suatu badan yang bernama ETSI (*European Telecommunication Standards Institute*) sebagai bagian dari pengembangan GSM Phase 2, yang terdapat pada dokumentasi GSM 03.40 dan GSM 03.38. Fitur SMS ini memungkinkan perangkat Stasiun Seluler Digital (*Digital Cellular Terminal*, seperti ponsel) untuk dapat

mengirim dan menerima pesan-pesan teks dengan panjang sampai dengan 160 karakter melalui jaringan GSM. (ETSI, 1996)

SMS dapat dikirimkan ke perangkat Stasiun Seluler Digital lainnya dalam beberapa detik selama berada jangkauan pelayanan GSM. Layanan SMS memberikan garansi SMS akan sampai pada tujuan meskipun perangkat yang dituju sedang tidak aktif yang dapat disebabkan karena sedang dalam kondisi mati atau berada di luar jangkauan layanan GSM. Jaringan SMS akan menyimpan sementara pesan yang belum terkirim, dan akan segera mengirimkan ke perangkat yang dituju setelah adanya tanda kehadiran dari perangkat di jaringan tersebut.

Jaringan GSM yang terintegrasi dengan layanan SMS memiliki tambahan subsistem, seperti gambar 2.1.



Gambar 2.1. Elemen pendukung SMS

2.1.1 Elemen-elemen SMS

a. *Short Messaging Entities*

Short messaging entity (SME) adalah suatu piranti yang dapat menerima atau mengirim pesan pendek. SME dapat berada dalam jaringan *fixed*, sebuah piranti bergerak, atau pusat layanan.

b. *Short Message Service Centre*

Short message service centre (SMSC) adalah kombinasi perangkat keras dan lunak yang bertanggung jawab memperkuat, menyimpan dan meneruskan pesan pendek antara SMS dan piranti bergerak. SMSC mentransfer pesan dalam format point to point pada sistem yang dilayani.

c. *SMS-Gateway dan SMS-Interworking Mobile Switching Center*

SMS gateway Mobile Switching Center (SMS-GMSC) adalah sebuah aplikasi MSC yang mampu menerima pesan singkat dari SMSC, menginterogasi *home location register* (HLR) untuk informasi routing, dan mengirimkan pesan pendek tersebut ke MSC dan piranti bergerak yang dituju.

d. *Home Location Register*

Home Location Register (HLR) adalah basis data yang digunakan untuk penyimpanan permanen, pengelolaan langganan dan profil layanan.

e. *Mobile Switching Center*

Mobile Switching Center (MSC) melakukan fungsi pensaklaran sistem dan mengendalikan panggilan ke dan dari sistem telepon dan data yang lain. MSC akan mengirimkan pesan pendek ke pelanggan tertentu melalui *base station* yang sesuai.

f. *Visitor Location Register*

Visitor Location Register (VLR) adalah basis data yang berisi informasi temporal mengenai pelanggan yang berasal dari suatu HLR yang *roaming* ke HLR lainnya.

g. *Base Station System*

Semua fungsi yang terkait dengan transmisi sinyal radio elektromagnetis antara MSC dan piranti bergerak dilakukan di *Base Station System* (BBS). BBS terdiri dari *Base Station Controllers* (BSCs). BSC dapat mengendalikan satu atau lebih BTS dan bertanggung jawab dalam pemberian sumber data.

2.1.2 Mekanisme *Store And Forward* pada SMS

SMS adalah data tipe *asynchronous message* yang pengiriman datanya dilakukan dengan mekanisme *protokol store and forward*. Hal ini berarti pengirim dan penerima SMS tidak perlu berada dalam status berhubungan (*connected/ online*) satu sama lain ketika akan saling bertukar pesan SMS. Pengiriman SMS secara *store and forward* berarti pengirim SMS menuliskan pesan dan nomor telepon tujuan dan kemudian mengirimkannya ke server SMS (*SMS-Center*) yang kemudian bertanggung jawab untuk mengirimkan pesan tersebut ke nomor telepon tujuan. Keuntungan mekanisme *store and forward* pada SMS adalah penerima tidak perlu dalam status *online* ketika ada SMS yang dikirimkan pesan kepadanya, karena pesan akan dikirim oleh pengirim ke SMSC yang kemudian dapat menunggu untuk meneruskan pesan tersebut ke penerima ketika ia siap dan dalam status *online* di lain waktu. Ketika pesan SMS telah terkirim dan diterima oleh SMSC, pengirim akan menerima pesan singkat (konfirmasi) bahwa pesan telah terkirim (*message sent*).

2.1.3 Cara Kerja SMS

Sekali pesan dikirim, pesan tersebut akan diterima dahulu oleh SMSC yang kemudian disampaikan pada nomor tujuan. Untuk melakukan proses ini SMSC

UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

mengirimkan sebuah SMS *request* ke HLR melalui *Signal Transfer Point* (STP) untuk menemukan pelanggan tujuan. Saat HLR menerima pesan tersebut maka HLR akan merespon ke SMSC dengan status pelanggan berupa:

1. *Inactive* atau *Active*
2. Letak pelanggan yang dimaksud (pelanggan tujuan).

Jika tidak aktif maka SMSC akan meng-*hold* pesan tersebut sampai pada periode tertentu. Saat pelanggan menyalakan *handset* maka akan terjadi *update location* pada HLR dan HLR akan mengirim status terhadap pesan yang belum terkirim. Jika aktif akan segera terkirim. SMSC menerima verifikasi jika pesan tersebut sudah diterima oleh nomor yang dituju dan mengkategorikan pesan tersebut sebagai sebuah "pesan terkirim" dan tidak akan melakukan percobaan pengiriman pesan lagi. Prinsip dasarnya adalah hanya ada satu *Short Message Service Center* yang menerjemahkan pesan untuk dikirimkan pada sebuah jaringan GSM. SMS dapat dikirimkan dan diterima bersamaan dengan *voice*, data dan *fax* menggunakan *channel* yang berbeda dengan SMS.

2.2 AT-Command

AT-*Command* merupakan perintah yang digunakan untuk berkomunikasi dengan modem. Ponsel pada dasarnya adalah sebuah modem, sehingga AT-*Command* pun berlaku untuk ponsel. AT-*Command* adalah bahasa yang sama dengan bahasa mesin, hanya untuk ponsel model berbeda..

Perintah-perintah AT-*Command* biasanya disediakan oleh vendor alat komunikasi. Contoh AT-*Command* pada SMS dilihat pada tabel 2.1 berikut.

Tabel 2.1 Beberapa perintah AT-Command

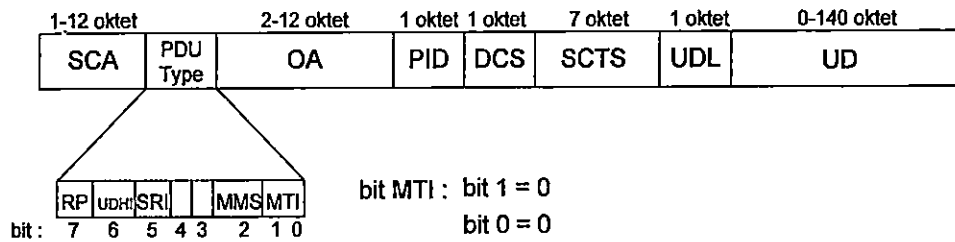
No	Perintah	Fungsi
1	AT+CMGL	Menampilkan daftar seluruh SMS
2	AT+CMGR	Membaca sebuah SMS
3	AT+CMGS	Mengirim SMS

2.3 Protokol Data Unit (PDU)

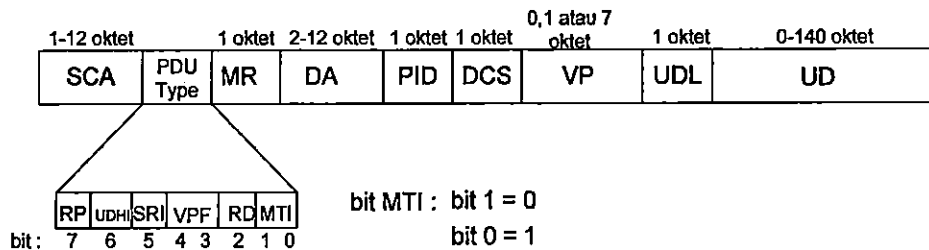
Dalam proses pengiriman atau penerimaan pesan pendek, data yang dikirim maupun diterima oleh stasiun bergerak menggunakan salah satu dari 2 mode yang ada, yaitu: mode teks, atau mode PDU (*Protocol Data Unit*) (Wavecom, 2000) [1].

Dalam mode PDU, pesan yang dikirim berupa informasi dalam bentuk data dengan beberapa kepala-kepala informasi. Hal ini akan memberikan kemudahan jika dalam pengiriman akan dilakukan kompresi data, atau akan dibentuk sistem penyandian data dari karakter dalam bentuk untaian bit-bit biner. PDU tidak hanya berisi pesan teks saja, tetapi terdapat beberapa meta-informasi yang lainnya, seperti nomor pengirim, nomor SMS Centre, waktu pengiriman, dan sebagainya[1].

Jenis PDU SMS yang akan digunakan adalah SMS-Penerimaan (*SMS-Deliver*) dan SMS-Pengiriman (*SMS-Submit*). SMS yang diterima akan memiliki format *SMS Deliver*. Sedang SMS yang ditulis atau akan dikirimkan akan memiliki format *SMS Submit*[2]. Struktur dari PDU penerimaan (*SMS-Deliver*) dan PDU pengiriman (*SMS-Submit*) dapat dilihat pada gambar di berikut[3].



Gambar 2.2 Struktur PDU Penerimaan (*SMS-Deliver*)

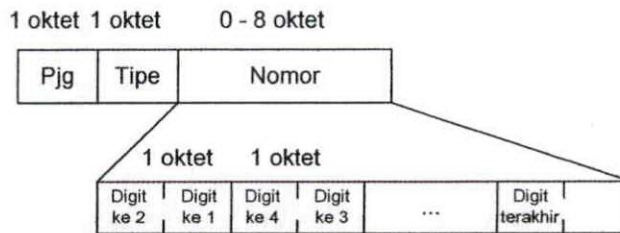


Gambar 2.3 Struktur PDU Pengiriman (*SMS-Submit*)

PDU penerimaan (*SMS-Deliver*) memiliki beberapa perbedaan dengan PDU pengiriman (*SMS-Submit*). Sebuah SMS dikatakan sebagai *SMS-Deliver* bila bit MTI dari SMS tersebut adalah "00". Sedangkan untuk *SMS-Submit* bit MTI dari SMS tersebut adalah "01". Hal ini dapat dilihat pada masing-masing gambar struktur PDU.

2.3.1 Service Centre Address (SCA)

Service Centre Address (alamat pusat layanan) merupakan bagian pertama pada struktur PDU baik PDU pengiriman maupun PDU penerimaan. Bagian ini berisi informasi-informasi mengenai *SMS Centre*. Hal ini dapat dilihat pada gambar elemen-elemen SCA bawah ini



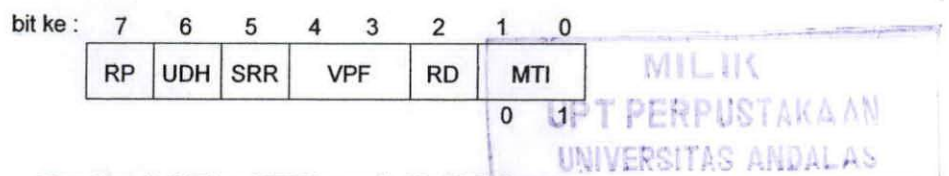
Gambar 2.4 Elemen-elemen penyusun *Service Centre Address*

Pada gambar terlihat bahwa sebuah SCA terdiri dari 3 elemen penyusun yaitu

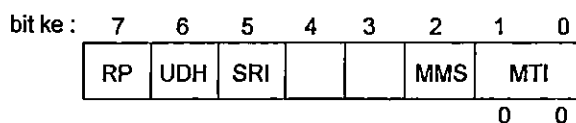
- a. Pjg, berisi panjang atau jumlah pasangan digit dari nomor SMSC (*service number*) yang digunakan[4]. Jumlah pasangan digit ini adalah 1 oktet.
- b. Tipe, berisi jenis pengkodean nomor SMSC. Pada bagian ini jenis pengkodean nomor SMSC terbagi atas 2 macam penomoran yaitu penomoran nasional dan penomoran internasional.
- c. Nomor, adalah nomor dari SMSC itu sendiri. Penulisan nomor SMSC ini dilakukan dengan membalikkan-balikkan tiap pasangan nomornya. Jika jumlah dari nomor SMSC itu ganjil, maka nomor terakhirnya dipasangkan dengan huruf F yang diletakkan didepannya.

2.3.2 Protocol Data Unit Type (PDU Type)

PDU *Type* ini berisi informasi mengenai jenis atau tipe dari PDU yang digunakan, apakah PDU *Deliver* atau PDU *Submit*. Elemen-elemen penyusun masing-masing PDU dapat dilihat pada gambar 2.5[3].



Gambar 2.5 Tipe PDU untuk *SMS Submit*



Gambar 2.6 Tipe PDU untuk *SMS Deliver*

Seperti terlihat pada gambar 2.5, PDU pengiriman memiliki beberapa perbedaan dengan PDU penerimaan.

2.3.2.1 *Reply Path (RP)*

Reply path atau jalur balasan berisi informasi mengenai keberadaan jalur balasan. Jika bit RP bernilai 0 berarti tidak ada *Service Centre* yang lain selain *Service Center operator* yang digunakan oleh penerima untuk membalas SMS. Jika bit RP ini bernilai 1 menandakan bahwa jalur balasan ada dan bisa digunakan untuk membalas SMS[5].

2.3.2.2 *User Data Header Indicator (UDHI)*

User Data Header Indicator berisi informasi mengenai indikator *header* pada sebuah SMS. Jika bit UDHI bernilai 0, berarti bagian *User Data* hanya mengandung pesan sedangkan jika UDHI bernilai 1 menandakan bahwa *User Data* memiliki header sebagai tambahan pesan. Penggunaan *Data Header* digunakan untuk pengiriman SMS yang berisi *ringtone* atau gambar [5].

2.3.2.3 *Status Report Indicator (SRI)*

Elemen ini berisi informasi mengenai status laporan pengiriman sebuah pesan. Elemen ini diset oleh SMSC. Jika SMSC menset bit SRI ini dengan nilai 0 maka laporan pengiriman tidak akan diberikan ke penerima. Sebaliknya penerima akan

menerima sebuah status laporan pengiriman pesan jika bit SRI ini diset dengan nilai 1 oleh SMSC.

2.3.2.4 Status Report Request (SRR)

Status Report Request berisi permintaan laporan pengiriman sebuah pesan. Bit SRR ini diset sendiri oleh pengguna ponsel. Jika bit bernilai 1, berarti pengirim meminta status pengiriman SMS dari *Service Center*, sedangkan jika bit bernilai 0 berarti tidak ada permintaan *Status Report* dari pengirim[5].

2.3.2.5 Validity Periode Format (VPF)

Validity Periode Format berisi informasi mengenai keberadaan dan format dari masa berlaku SMS. Parameter ini ditentukan oleh 2 buah bit dengan konfigurasi seperti yang terdapat pada tabel 2.2 [5].

Tabel 2.2 Konfigurasi bit *Validity Periode Format*

Bit ke-4	Bit ke-3	Keterangan
0	0	Tidak ada informasi tentang masa berlaku SMS tersebut.
0	1	Tidak digunakan
1	0	Informasi masa berlaku SMS dalam bentuk integer (relatif)
1	1	Informasi masa berlaku SMS dalam semi-oktet (absolut)

2.3.2.6 Reject Duplicate (RD)

Reject Duplicate berisi informasi mengenai penolakan SMS duplikat. Bit ini berfungsi untuk memberitahu *Service Center* tindakan yang akan diambil jika ada SMS yang memiliki MR (*Message Reference*) dan nomor penerima yang sama yang

berasal dari pengirim yang sama jika SMS sebelumnya masih berada/tersimpan di *Service Center*. Jika bit ini bernilai 1 maka *SMS Centre* akan menolak setiap pesan yang berasal dari pengirim yang sama dengan isi pesan yang sama selama pesan tersebut masih tersimpan di *SMS Centre*. Namun jika bit ini bernilai 0 maka *SMS Centre* akan mengirimkan semua pesan yang tersimpan di *SMS Centre* kepada penerima walaupun pengirim dan isi pesannya adalah sama.

2.3.2.7 *More Message to Send* (MMS)

More Message to Send berisi informasi bahwa masih ada SMS lain yang akan dikirimkan sebagai sambungannya. Elemen ini diset oleh SMSC. Jika bit ini bernilai 0 berarti masih ada SMS lain yang akan dikirimkan sebagai lanjutannya. Sebaliknya jika bit ini bernilai 1 maka berarti tidak ada SMS lain yang akan diterima oleh penerima sebagai lanjutannya.

2.3.2.8 *Message Type Indicator* (MTI)

Message Type Indicator berisi informasi mengenai indikator jenis SMS yang digunakan. Bagian ini terdiri dari 2 bit. Bit-bit ini akan menentukan apakah sebuah SMS digunakan sebagai *SMS Deliver* atau sebagai *SMS Submit*. Adapun konfigurasi dari bit MTI adalah sebagai berikut [5].

Tabel 2.3 Konfigurasi bit MTI

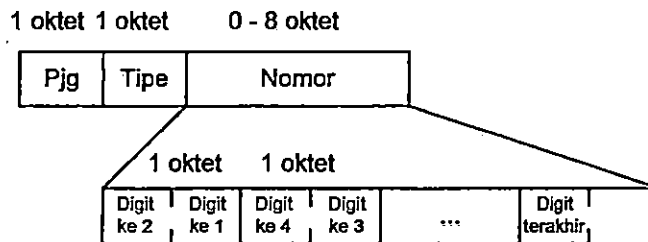
Bit 1	Bit 0	Protokol yang digunakan
0	0	SMS- <i>DELIVER</i>
0	0	SMS- <i>DELIVER REPORT</i>
0	1	SMS- <i>SUBMIT</i>
0	1	SMS- <i>SUBMIT REPORT</i>
1	0	SMS- <i>STATUS REPORT</i>
1	0	SMS- <i>COMMAND</i>
1	1	Tidak dipergunakan

2.3.3 Message Reference (MR)

Message Reference atau referensi pesan berisi informasi mengenai referensi terhadap jumlah SMS yang telah dikirimkan dari *Mobile Station* kepada *Service Center*. Informasi ini tersimpan dalam elemen MR sepanjang 1 oktet.

2.3.4 Originator Address (OA) dan Destination Address (DA)

OA dan DA memiliki konfigurasi yang sama dengan SCA. Hal ini dapat dilihat dari konfigurasi gambar 2.7 berikut [5].



Gambar 2.7 Format PDU untuk nomor penerima dan nomor pengirim

Elemen penyusun OA dan DA terdiri dari 3 bagian yaitu :

- a. Pjg. Pada OA bit berisi informasi panjang dari nomor pengirim, sedangkan pada DA berisi informasi mengenai panjang nomor penerima.
- b. Tipe. Seperti pada SCA bit berisi informasi apakah menggunakan format nasional atau format internasional
- c. Nomor. Pada OA bit berisi mengenai nomor pengirim pesan sedangkan pada DA berisi informasi mengenai nomor penerima pesan. Isi dari nomor ini juga dipasang terbalik-balik serta jika nomor pengirim atau penerima berjumlah ganjil maka nomor terakhir akan dipasangkan dengan huruf F pada bagian depannya.

2.3.5 Protokol Identifier (PID)

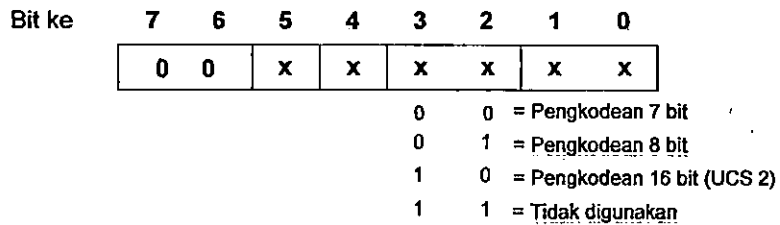
PID (*Protocol Identifier*) memiliki panjang 1 oktet. Parameter ini akan memberikan informasi tentang protokol yang digunakan oleh SMS. *Protocol Identifier* ini akan memberikan informasi tentang cara *Service Center* memperlakukan SMS tersebut. *Service Center* dapat menolak SMS yang memiliki protokol *identifier* yang tidak didukungnya atau mengandung nilai yang tidak dipergunakan [5].

Umumnya setiap SMS yang digunakan saat ini menggunakan Protokol *Identifier* dengan konfigurasi "00" yang menandakan bahwa SMS akan diperlakukan sebagai SMS text biasa [5].

2.3.6 Data Coding Sheme (DCS)

Data Coding Sheme berisi informasi mengenai skema pengkodean SMS. Skema pengkodean disimpan oleh elemen DCS. Elemen DCS ini akan memberikan informasi tentang jenis pengkodean yang digunakan pada pesan text yang terdapat

pada bagian *User Data* (UD). Selain itu DCS ini juga akan menentukan kelas dari pesan tersebut. Panjang dari DCS ini adalah 1 oktet. Struktur data untuk elemen DCS dapat dilihat pada gambar 2.9 [5].



Gambar 2.8 Konfigurasi bit *Data Coding Scheme* (DCS)

Pengkodean elemen DCS yang umum digunakan adalah pengkodean DCS dengan menggunakan *coding group* "00xx" yang dikenal dengan nama "*General Data Coding*". Pengkodean DCS "*General Data Coding*" dapat dilihat pada tabel dibawah ini. Konfigurasi bit lainnya dapat dilihat pada tabel 2.4 [5].

Tabel 2.4 Konfigurasi bit ke-5 DCS

Bit 5	Keterangan
0	Teks tidak dikompres
1	Teks dikompres

Tabel 2.5 Konfigurasi bit ke-4 DCS

Bit 4	Keterangan
0	Bit 1 dan 0 tidak digunakan dan tidak mempunyai arti
1	Bit 1 dan 0 memiliki kelas Message

Tabel 2.6 Konfigurasi bit ke-1 dan bit 0 DCS

Bit 1	Bit 0	Message Class	Keterangan
0	0	Class 0	Tampilkan segera
0	1	Class 1	Tergantung ME
1	0	Class 2	Tergantung SIM
1	1	Class 3	Tergantung TE

Konfigurasi untuk bit ke-3 dan bit ke-2 dapat dilihat pada tabel 2.7 yang merupakan bit untuk menentukan pengkodean yang digunakan dalam elemen UD.

2.3.7 Indikator Waktu *Service Centre* (SCTS)

Indikator waktu *Service Center* dikenal dengan nama SCTS (*Service Centre Time Stamp*). Elemen SCTS pada PDU memberikan informasi tentang waktu SMS tersebut diterima oleh *Service Center*. Panjang informasi SCTS ini adalah 7 oktet dimana oktet tersebut menyimpan informasi tahun, bulan, tanggal, jam, menit, detik dan zona waktu.

2.3.8 Masa Berlaku SMS (*Validity Period*)

Masa berlaku SMS disimpan dalam elemen VP (*Validity Period*) apabila bit VPF pada elemen konfigurasi PDU tidak bernilai "00". VP berfungsi untuk menentukan seberapa lama *Service Center* menyimpan SMS yang dikirimkan apabila SMS tersebut belum bisa diteruskan kepada penerimanya. Ada dua jenis format VP yaitu format relatif dan format absolut [5].

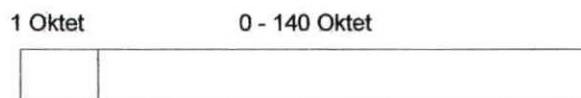
Format relatif menggunakan angka integer sebagai penghitung waktu. Jangka waktu keabsahan suatu SMS dihitung dari waktu SMS diterima oleh *Service Center*.

Format absolut menggunakan tanggal untuk menentukan masa berlaku SMS. Dengan demikian apabila tanggal yang terdapat dalam VP telah berlalu, maka SMS tersebut dinyatakan sudah kadaluarsa dan tidak akan dikirimkan lagi ke penerimanya.

Masa berlaku dari sebuah SMS tidak hanya ditentukan oleh nilai VP semata tapi juga dukungan dari *Service Center*. Tidak semua *Service Center* yang memberikan toleransi *Validity Period* untuk jangka waktu yang lama.

2.3.9 *User Data Length (UDL) dan User Data (UD)*

UDL berisi Informasi tentang panjang yang akan dikirimkan berupa jumlah oktet yang terdapat pada UD. Sedangkan UD berisi informasi isi dari pesan yang akan dikirimkan. Panjang UD maksimum untuk sebuah SMS adalah 140 oktet yang artinya dapat membawa 160 karakter jika menggunakan default alphabet (7-bit/karakter) dan 70 karakter jika menggunakan UCS2 (16 bit per karakter). Struktur data UDL dan UD dapat dilihat pada gambar 2.12 berikut [5].



Gambar 2.9 Struktur PDU *User Data Length (UDL)* dan *User Data (UD)*

2.3.10 Contoh Penggunaan PDU Pengiriman (SMS – *Submit*)

Contoh penggunaan struktur PDU pengiriman, sebuah SMS *submit* dapat berisi informasi sebagai berikut :

07 91 2658050000F0 11 00 0C 91 265836164900 00 00 FF 04 C830FB0D

Keterangan :

- 07 adalah panjang atau jumlah pasangan digit dari nomor SMSC (*service number*) yang digunakan, dalam hal ini adalah 7 pasangan (14 digit berikutnya)
- 91 adalah jenis nomor SMSC. Angka 91 menandakan format nomor internasional (misal +6281xxx). Untuk 081xxx menggunakan angka 81.
- 2658050000F0 adalah nomor SMSC yang digunakan. Karena jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah +6285500000 (IM3)
- 11 adalah oktet pertama untuk PDU SMS untuk dikirim (*SMS Submit*).
- 00 adalah TP-Message-Reference. Diisi "00" agar diisi otomatis oleh handphone.
- 0C adalah panjang digit dari nomor penerima (0C hex = 12 desimal)
- 91 adalah jenis nomor penerima (sama dengan jenis nomor SMSC)
- 265836164900 ,Nomor penerima SMS, yang jika diterjemahkan adalah +628563619400
- 00 , Pengenal protokol, dalam hal ini adalah 0
- 00 , Skema pengkodean SMS, juga bernilai 0
- FF , Validitas waktu. FF berarti maksimum.
- 04, Panjang dari pesan SMS, dalam hal ini adalah 4 huruf (mode 7-bit).
- C830FB0D, Pesan SMS dalam mode 7-bit. Jika diterjemahkan kedalam 8-bit, lalu dirubah ke ASCII, maka didapat pesan 'Halo'

2.4 Pengkodean SMS

SMS memiliki beberapa jenis pengkodean terhadap isi SMS yaitu pengkodean 7-bit, 8-bit dan 16-bit. Penggunaan ketiga jenis pengkodean tersebut tergantung pada keperluan. Pengkodean 7-bit digunakan untuk pengiriman SMS teks biasa dimana 128 karakter yang dapat ditampung dan proses pengkodean yang sedikit rumit.. Pengkodean 8-bit lebih banyak digunakan untuk keperluan pengiriman data berupa gambar dan nada *ringtone*. Pengkodean 16-bit biasa digunakan untuk pengiriman pesan teks yang lebih kompleks yaitu teks yang menggunakan karakter non-Latin seperti huruf arab, karakter kanji, karakter Cina dan lain sebagainya. Hal ini disebabkan banyaknya bit yang digunakan untuk merepresentasikan satu karakter yaitu 16 bit sehingga terdapat 2^{16} karakter atau sebanyak 65536 karakter. Selain untuk pengiriman teks yang kompleks, pengkodean 16 bit ini juga digunakan untuk pengiriman SMS yang berkedip atau yang biasa disebut SMS *Flash*.

Pengkodean dari pesan text SMS dilakukan berdasarkan konfigurasi bit DCS (*Data Coding Scheme*) yang terdapat pada konfigurasi bit SMS baik SMS-*DELIVER* maupun SMS-*SUBMIT*.

Komputer atau mikroprosesor pada umumnya memiliki bentuk data berupa byte yang terdiri dari 8-bit. Sedangkan pengkodean 7-bit hanya memiliki 7-bit untuk tiap karakternya. Oleh karena itu diperlukan proses konversi data 7-bit menjadi data 8-bit dalam proses pengiriman dan proses konversi data 8-bit menjadi data 7-bit pada proses penerimaan.

Pengkodean SMS ditentukan dari konfigurasi bit 3 dan 2 dari DCS dengan konfigurasi sebagai berikut Konfigurasi ini berlaku bila bit 7 dan 6 bernilai 00. Penggunaan bit 3 dan bit 2 DCS ini dapat dilihat pada tabel 2.7 berikut ini.

Tabel 2.7 Konfigurasi bit DCS dan pengkodean yang digunakan

Bit 3	Bit 2	Jenis pengkodean	Panjang text/data setiap pesan
0	0	Default alphabet (karakter 7-bit)	160 karakter
0	1	8-bit data	140 byte
1	0	UCS2 (16bit)	70 karakter <i>Unicode</i>
1	1	<i>Reserved</i>	-

2.5 Sensor Ultrasonik

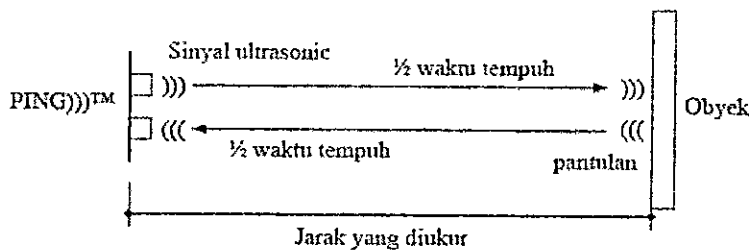
Sensor merupakan komponen yang di gunakan untuk mendeteksi dan sering berfungsi untuk mengukur magnitude. Dalam aplikasinya sensor tidak dapat kita pisahkan dari sistem sensor yang akan mengolah lebih lanjut besaran elektronik yang dihasilkan oleh sensor sehingga dapat kita baca atau hitung dalam bentuk besaran lainnya.

Sensor yang digunakan adalah Modul PING)))™. Sensor ping mengukur jarak obyek dengan cara memancarkan gelombang ultrasonik (40 kHz) dan kemudian menangkap kembali pantulan gelombangnya. Sensor ping memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger*). Gelombang ultrasonik ini melalui udara dengan kecepatan kurang lebih 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. Ping akan

mengeluarkan pulsa *output hig* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi, sensor ping akan membuat *output low* pada pin SIG [6].

Untuk perhitungan sensor ping dapat menggunakan persamaan berikut :

$$\text{Jarak benda} = \text{waktu pantul} \times 34,4 \quad (\text{dalam cm}) \quad (2.1)$$



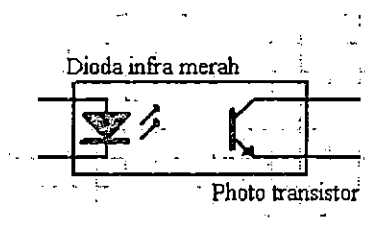
Gambar 2.10. Ilustrasi cara kerja sensor PING)))™

2.6 Sensor Kecepatan

Sensor kecepatan yang digunakan adalah photo transistor dan dioda infra infra. Dioda infra merah merupakan komponen yang berfungsi untuk memancarkan frekuensi radiasi yang bekerja dibawah tingkat sensitivitas mata manusia dan mirip dengan dioda led hanya saja pada dioda led jika diberi tegangan maka led akan menyala, kalau dioda infra merah kebalikkannya. Dioda infra merah sering dipakai untuk berkomunikasi dengan photo transistor. Infra merah merupakan gelombang elektromagnetik, yang tidak bisa diamati secara langsung dengan menggunakan mata telanjang.

Photo transistor adalah komponen semi konduktor dan sama dengan transistor bipolar biasa, bedanya tidak terdapat terminal kaki basis. Jadi sebagai kaki

basisnya ialah terdapat berupa lensa dipermukaannya. Jika diberikan berupa cahaya terutama cahaya infra merah. Komponen ini akan aktif yaitu akan timbul arus basis dan inilah sebagai pengganti arus input.



Gambar 2.11. Simbol dioda infra merah dan photo transistor

2.7 Liquid Crystal Display (LCD)

Layar LCD merupakan media penampil data yang sangat efektif dalam suatu sistem elektronik. Agar sebuah pesan atau gambar dapat tampil pada layar LCD, diperlukan sebuah rangkaian pengatur *scanning* dan pembangkit tegangan sinus.

Modul LCD yang digunakan modul LCD M1632. Modul ini merupakan Modul LCD Matriks dengan konfigurasi 16 karekater dan 2 baris dengan setiap karakternya dibentuk oleh 8 baris pixel dan 5 kolom pixel (1 baris pixel terakhir adalah kursor).

Tabel 2.8. Konfigurasi pin- pin LCD

No	Nama	Fungsi
1	Vcc	+ 5V
2	GND	0 V
3	VEE	Tegangan kontras LCD

4	RS	<i>Register Select, 0 = register perintah, 1 = register data</i>
5	R/W	<i>1 = read, 0 = write</i>
6	E	<i>Enable clock LCD</i>
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	Anode	Tegangan positif <i>backlight</i>
16	Katode	Tegangan negatif <i>backlight</i>

Sumber . Belajar Sendiri Mikrokontroler AVR Seri ATmega8535[7]

2.8 Mikrokontroler AT89C52

Mikrokontroler AT89C52 merupakan salah satu mikrokontroler keluaran dari ATMEL yang termasuk dalam keluarga mikrokontroler MCS-51.

Adapun spesifikasi teknis yang di miliki oleh Pengendalian mikro AT89C52 adalah:

1. 8 Kbytes *Reprogrammable ROM*
2. Dapat di program sebanyak 1000 kali
3. Memiliki 256 x 8 intemal RAM
4. 32 *Progmmable I/O Lines*

5. 3 buah 16 bit *Timer/Counters*

6. 8 buah sumber *interrupt*

2.8.1 Konsep Dasar Mikrokontroler

Suatu mikroprosesor adalah bagian CPU (*Central Processing Unit*) dari sebuah komputer, tanpa memori, I/O, dan peripheral dibutuhkan suatu sistem lengkap.

Diantara keuntungan pemakaian mikrokontroler dibandingkan dengan mikroprosesor adalah pada mikrokontroler sudah RAM dan peralatan I/O pendukung sehingga tidak perlu menambahkannya lagi.

Mikrokontroler merupakan sebuah mikroprosesor yang dikombinasikan dengan I/O dan memori (RAM/ROM) yang dalam sebuah *single chip*. Mikrokontroler menggunakan EPROM sebagai penyimpanan program. Mikroprosesor adalah bagian CPU (*Central Processing Unit*) dari suatu komputer. Mikroprosesor membutuhkan perangkat pendukung untuk dapat bekerja yaitu RAM (*Random Acces Memory*), ROM (*Read Only Memory*) dan I/O (*Input Output*).

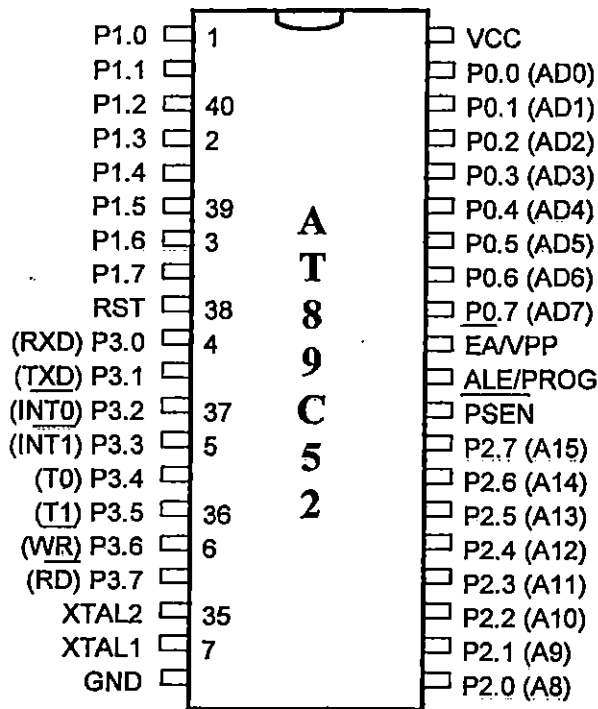
2.8.2 Bahasa *Assembly* MCS-51

Sebuah mikrokontroler tidak akan bekerja bila tidak diberikan program kepadanya. Program tersebut memberitahu mikrokontroler apa yang harus dilakukan. Sebuah mikrokontroler yang telah bekerja baik dengan suatu program, tidak akan bekerja lagi bila programnya diganti [8].

Instruksi-instruksi perangkat lunak berbeda untuk masing-masing jenis mikrokontroler. Instruksi-instruksi ini hanya bisa dipahami oleh jenis mikrokontroler

yang bersangkutan. Sebuah mikrokontroler tidak dapat memahami instruksi-instruksi yang berlaku pada mikrokontroler lain kecuali yang kompatibel.

2.8.3 Susunan Pin-pin Mikrokontroler AT89C52



Gambar 2.12. Susunan Pin (Kaki) Mikrokontroler AT89C52

Susunan pin-pin mikrokontroler AT89C52 mempunyai fungsi sebagai berikut:

- Pin 1 sampai 8 (port 1) merupakan *port paralel* 8-bit, *port* dapat digunakan untuk mengirim atau menerima data (*birectional*).
- Pin 9 adalah masukan reset, dengan memberikan perubahan pulsa dari rendah ke tinggi (aktif tinggi) akan me-reset mikrokontoler.
- Pin 10 sampai 17 (*port 3*) adalah *port paralel* 8-bit dua arah yang memiliki fungsi pengganti, fungsi pengganti meliputi TxD (*Transmit Data*), RxD (*Recceive Data*),

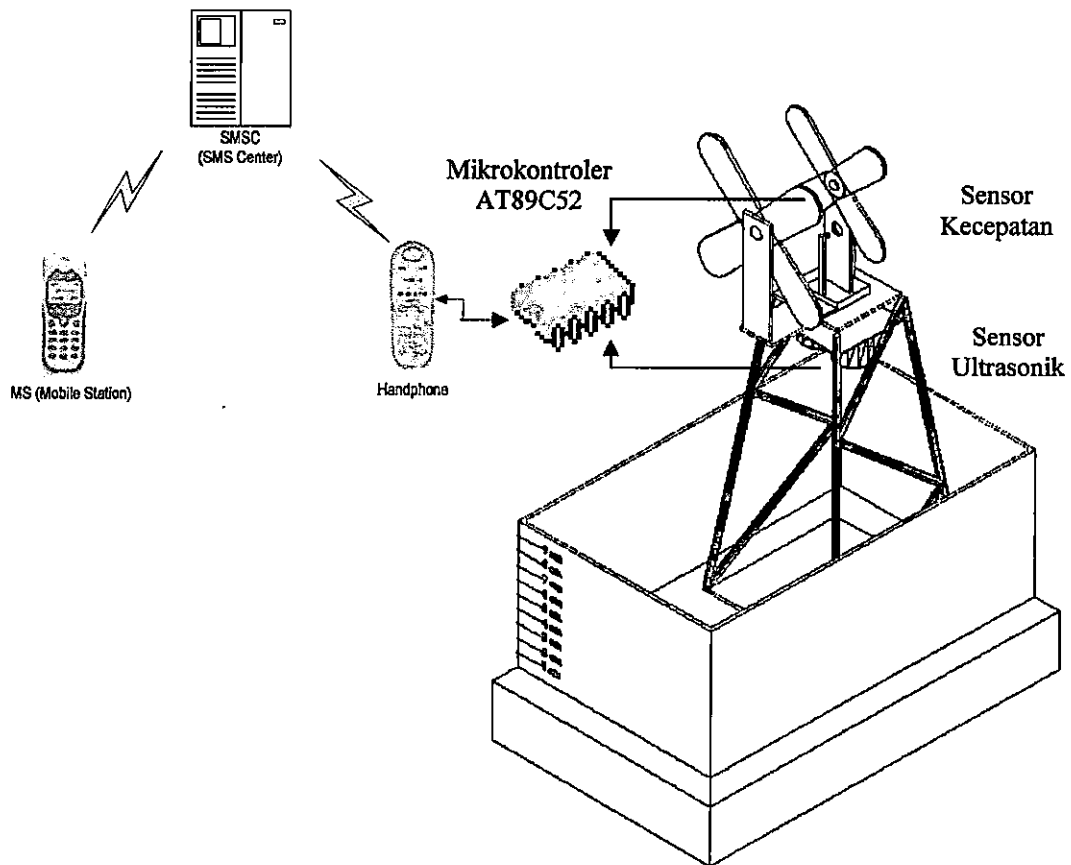
Int 0 (*Interrupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), WR (*Write*), dan RD (*Read*), bila fungsi pengganti tidak dipakai dapat digunakan sebagai *port paralel* serba guna.

- d. Pin 18 (XTAL 1) adalah pin masukan ke rangkaian *osilator internal*, sebuah kristal atau sumber osilator lain diluar 89C52 dapat digunakan.
- e. Pin 19 (XTAL 2) adalah pin keluaran ke rangkaian osilator internal, pin ini dipakai bila menggunakan osilator kristal.
- f. Pin 20 (*Ground*) dihubungkan ke *ground* dari catu daya.
- g. Pin 21 sampai 28 (*port 2*) adalah *port paralel 8-bit* dua arah (*bidirectional*) *port* mengirimkan byte alamat bila dilakukan pengaksesan memori luar.
- h. Pin 29 adalah pin $\overline{\text{PSEN}}$ (*Program Store Enable*) yang merupakan sinyal pengontrol yang memperbolehkan program memori eksternal masuk kedalam bus selama proses pemberian atau pengambilan instruksi (*Fetching*).
- i. Pin 30 adalah pin ALE (*Address Latch Enable*) yang digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
- j. Pin 31 ($\overline{\text{EA}}$), bila pin diberi logika tinggi (H), mikrokontroler akan melaksanakan instruksi dari memori program dalam (*internal*). Bila diberi logika rendah (L), mikrokontroler akan melaksanakan instruksi dari memori luar (*eksternal*).
- k. Pin 32 sampai 39 (*port 0*) merupakan *port paralel 8-bit* dua arah, bila digunakan untuk mengakses memori luar, ini memultipleks alamat memori dengan data.
- l. Pin 40 (Vcc) dihubungkan ke Vcc (+ 5 volt) [8].

BAB III

PERANCANGAN SISTEM

Dalam perancangan sistem dibagi menjadi 2 bagian yaitu perancangan perangkat keras (rangkaiannya) dan perancangan perangkat lunak (program). Perangkat keras terdiri dari beberapa sub sistem yang menyusun sistem ini menjadi sebuah sistem tunggal. Program yang dirancang adalah suatu perangkat lunak yang dapat melakukan pengambilan data yang akan diproses dalam bentuk isi SMS yang akan dikirimkan ke ponsel *master*.



Gambar 3.1 Skema rancangan sistem

3.1 Perancangan Perangkat Keras

Perangkat keras yang digunakan terdiri dari beberapa sub sistem yaitu telepon *seluler* (ponsel), rangkaian sensor, dan rangkaian pengendali (mikrokontroler).

3.1.1 Ponsel

Ponsel yang digunakan adalah merek Siemens seri C35/C45. Ponsel ini memiliki karakteristik yang sangat kompatibel untuk melakukan komunikasi secara serial serta mendukung (*support*) untuk perintah-perintah AT Command.

Komunikasi antara ponsel dengan mikrokontroler adalah komunikasi serial jenis USART. Pada komunikasi hanya ada 1 bit yang dapat dikirimkan atau diterima dalam satu waktu dan bit-bit tersebut dikirimkan secara berurutan. Banyaknya bit yang dapat dikirimkan dalam satu satuan waktu (biasanya dalam detik) dinyatakan dengan *baud rate* dalam satuan bps (*bit per second*). Pada perancangan nilai *baud rate* yang digunakan adalah sebesar 19.200 bps. Artinya dalam satu detik banyaknya bit yang dapat dikirimkan atau diterima adalah sebesar 19.200 bit. Nilai ini adalah nilai yang banyak mendukung pada kebanyakan merek ponsel.

Untuk komunikasi menghubungkan antara ponsel dengan mikrokontroler dapat langsung dihubungkan. Walaupun mikrokontroler dan ponsel memiliki level tegangan yang tidak sama, dimana mikrokontroler memiliki level tegangan TTL (*Transistor Transistor Logic*) dimana bit 0 didefinisikan dengan tegangan 0 volt dan bit 1 didefinisikan dengan tegangan 5 volt sedangkan ponsel memiliki level tegangan positif bipolar dimana bit 0 didefinisikan dengan tegangan 0 volt sedangkan bit 1 didefinisikan dengan tegangan 2.7 volt. Secara teori nilai 2.7 volt

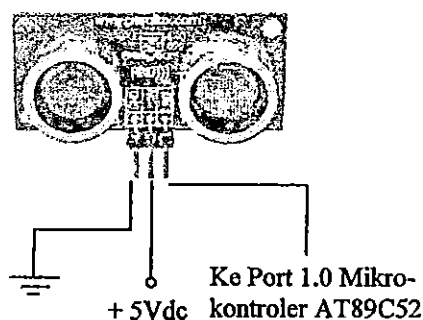
masih dihitung sebagai level tegangan TTL sehingga antara ponsel dengan mikrokontroler dapat langsung dihubungkan.

3.1.2 Rangkaian Sensor

Dalam rangkaian sensor digunakan 2 sensor yaitu sensor ultra ultrasonic yang digunakan untuk mengambil data dari perubahan level ketinggian air dan sensor kecepatan untuk menghitung data kecepatan angin

3.1.2.1 Rangkaian Sensor Ultrasonik

Rangkaian sensor Ping yang digunakan berbentuk modul buatan Parallax. Modul sensor Ping memiliki 3 buah pin. Hubungan modul sensor Ping seperti gambar berikut.

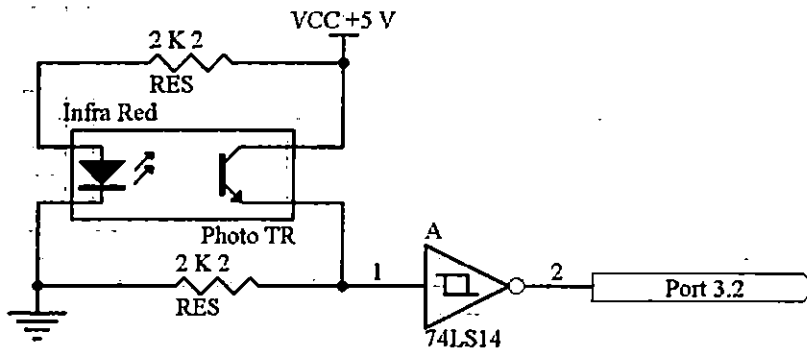


Gambar 3.2 Hubungan modul sensor ping dengan MC AT89C52

Rangkaian ini terdiri dari 2 sensor, sensor pengirim dan sensor penerima. Rangkaian bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor ini menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar pengindraannya. Perbedaan waktu antara gelombang suara dipancarkan dengan ditangkapnya kembali gelombang suara tersebut adalah berbanding lurus dengan jarak atau tinggi objek yang memantulkannya.

3.1.2.2 Rangkaian Sensor Kecepatan

Rangkaian sensor berfungsi untuk mendeteksi hasil pengukuran kecepatan angin yang diperoleh dari putaran baling-baling. Sensor kecepatan yang digunakan dari dioda infra merah dan photo transistor. Keluaran dari rangkaian sensor ini diambil dari pin emiter photo transistor. Rangkaian sensor kecepatan dapat dilihat pada gambar 3.3 berikut.



Gambar 3.3 Rangkaian sensor kecepatan

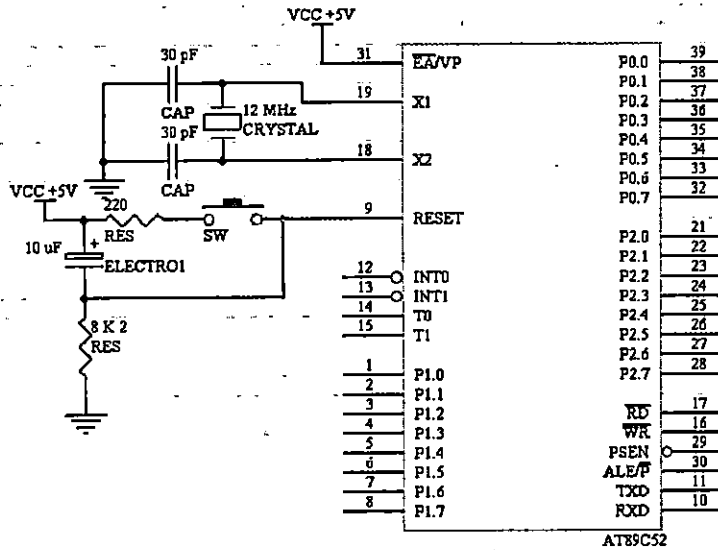
Tegangan keluaran pin emiter photo transistor diinputkan sebuah IC LS7413 Inverter schmitt trigger berfungsi untuk menyempurnakan keluaran logika 0 atau 1 yang dibentuk oleh sebuah rangkaian penghasil pulsa sekaligus menghilangkan noise pada gelombang outputnya sehingga pulsa *clock* yang dihasilkan benar-benar berbentuk gelombang berbentuk simetris. Pulsa *clock* yang dihasilkan inverter schmitt trigger dihubungkan ke input ke langsung di inputkan pada pin P3.2 mikrokontroler AT89C52. Untuk menghitung kecepatan angin digunakan sebuah *counter*. Dalam sistem ini rangkaian *counter* tidak digunakan tetapi perhitungannya dilakukan dengan perangkat lunak atau berupa program *counter*. Program *counter* ini dimaksud agar mengefesienkan rangkaian dan memaksimalkan penggunaan dari mikrokontroler itu sendiri.

3.1.3 Rangkaian Sistem Mikrokontroler

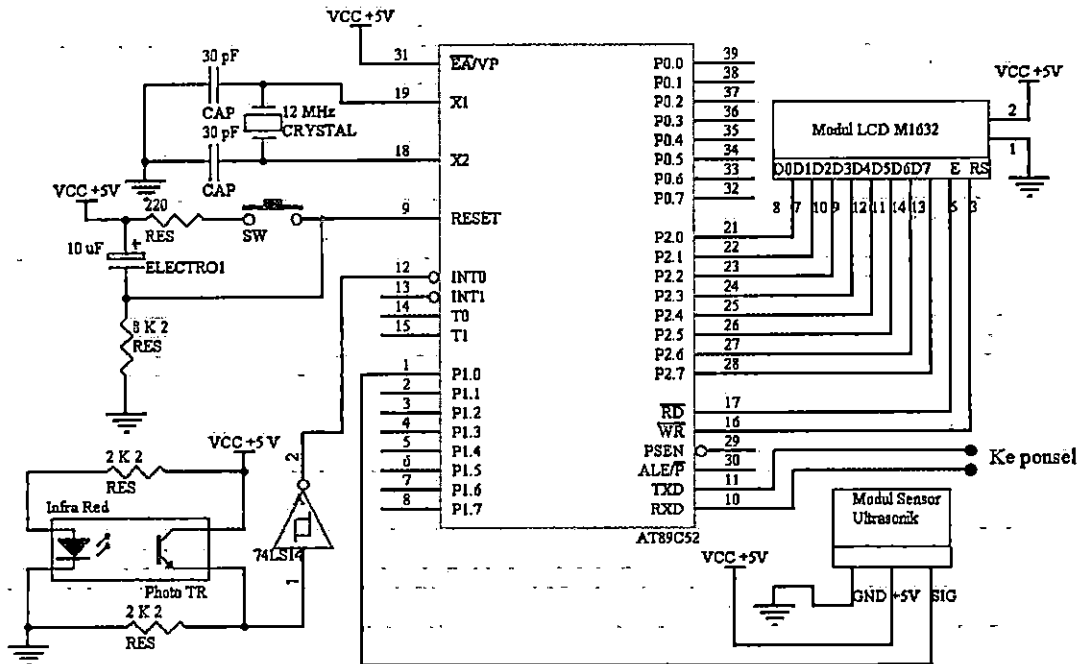
Mikrokontroler AT89C52 digunakan sebagai pengendali sistem secara keseluruhan. Mikrokontroler akan melakukan pengendalian mulai dari pengambilan data dari pengecekan level ketinggian air dan kecepatan angin, dan akan memproses data dari kedua sensor, selanjutnya melakukan proses pengiriman SMS dan menampilkan hasil yang didapat dari kedua sensor yang digunakan.

Level ketinggian air tidak selalu berubah sebanding dengan adanya perubahan kecepatan angin. Untuk mendapatkan perubahan level ketinggian air tidak perlu mendapatkan perubahan kecepatan angin yang signifikan. Maka dalam perancangan sistem ini lebih difokuskan pada perubahan level ketinggian air dikarenakan data yang diperlukan untuk pengiriman data SMS adalah perubahan level ketinggian air. Seandainya dalam sistem ini berpatokan terhadap perubahan kecepatan angin maka pengiriman data SMS akan dilakukan setiap saat sesuai dengan perubahan dari kecepatan angin, dikarenakan kecepatan angin tiap saat dapat berubah atau tidak konstan.

Pengendali sistem ini terdiri dari rangkaian sistem minimum mikrokontroler AT89C52, modul sensor ultrasonic, rangkaian sensor kecepatan dan modul LCD. Rangkaian sensor ultrasonic yang digunakan adalah sensor ping yang berbentuk modul buatan Parallax. Rangkaian ini berguna sebagai sensor untuk mengukur perubahan level ketinggian air, dan untuk rangkaian sensor kecepatan digunakan dioda infra merah dan photo transistor untuk mendeteksi perubahan kecepatan angin. Sedangkan untuk LCD digunakan modul LCD M1632.



Gambar 3.4. Rangkaian sistem minimum mikrokontroler AT89C52



Gambar 3.5 Rangkaian lengkap sistem

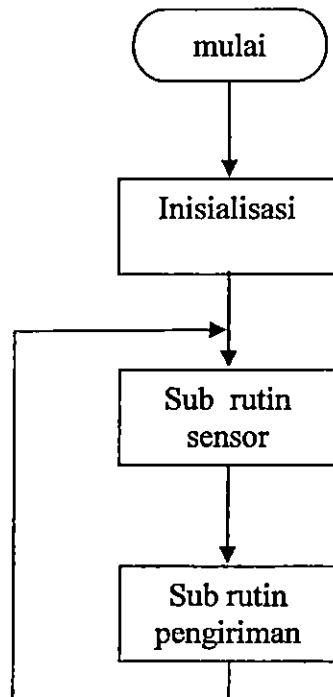
3.2 Perancangan Perangkat Lunak

Perangkat lunak yang dipakai adalah program yang disusun dengan bahasa pemrograman ALDS (*Asembly Language Definition System*). Perancangan perangkat lunak ini diharapkan mampu melakukan untuk :

- a. Membaca setiap perubahan level ketinggian air setiap waktu
- b. Membaca kecepatan angin dalam rentang waktu 60 detik.
- c. Menterjemahkan data dari level ketinggian air dan kecepatan angin
- d. Melakukan pengiriman SMS jika level ketinggian air berubah

3.3 Algoritma Perangkat Lunak

Perancangan perangkat lunak terdiri dari beberapa algoritma dengan mengacu kepada tiap-tiap subsistem yang menyusun sistem ini mulai dari algoritma inisialisasi, pembacaan nilai sensor dan sampai proses pengiriman SMS.



Gambar 3.6 Diagram alir perangkat lunak

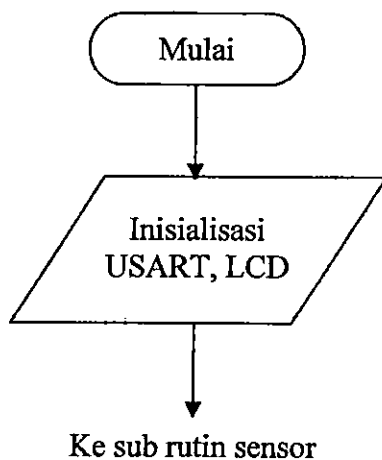
3.3.1 Sub Rutin Inisialisasi

Proses inisialisasi dilakukan pada komunikasi serial antara mikrokontroler dengan ponsel. Proses pengiriman dan penerimaan pesan menggunakan

komunikasi serial USART (*Universal Synchronous Asynchronous Receiver Transmitter*)

Pada komunikasi serial ini digunakan nilai *baudrate* sebesar 19.200 bps. Nilai ini adalah nilai yang sangat mendukung bagi ponsel untuk melakukan komunikasi serial baik dengan mikrokontroler ataupun dengan komputer.

Untuk menampilkan data pada LCD, diperlukan juga proses inisialisasi antara mikrokonteler dengan LCD.



Gambar 3.7 Algoritma sub sistem inisialisasi

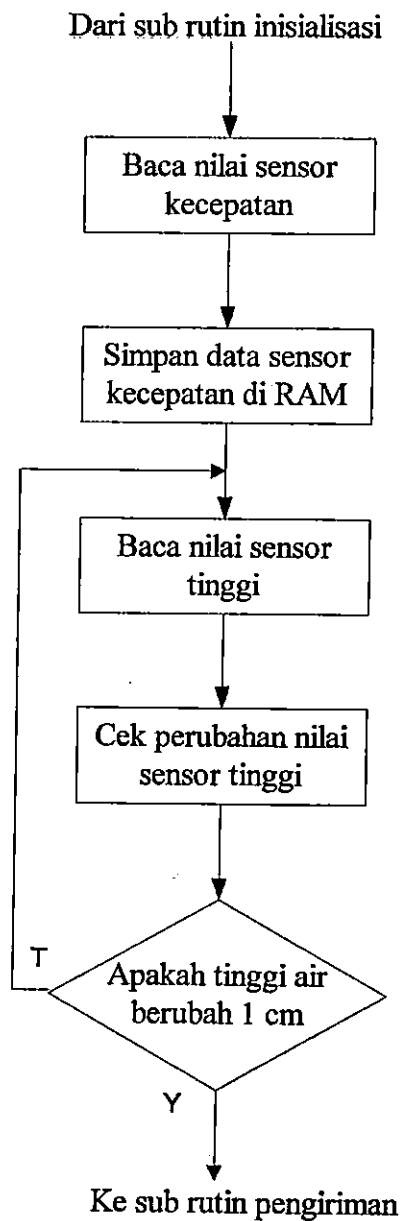
3.3.2 Sub Rutin Sensor

Sub rutin sensor dirancang untuk membaca level ketinggian air dan kecepatan angin. Ketika alat pertama kali diaktifkan, alat akan mendeteksi level ketinggian air dan kecepatan angin, mengolah data tersebut dan mengirimkan SMS ke ponsel master.

Untuk menghitung kecepatan angin, kita menggunakan *counter* berupa perangkat lunak (*software*). Perangkat lunak untuk menghitung kecepatan angin dihitung dalam bentuk rpm (*rotation per menit*). Algoritma dalam perangkat lunak selama 60 detik akan menghitung putaran baling-baling, data hitungan tersebut

akan disimpan pada RAM dan conter perangkat lunak akan mulai menghitung kecepatan angin selanjutnya mulai dari nol.

Perubahan dari level ketinggian air maka algoritma program akan mengirimkan SMS ke posel master yang mana data yang dikirimkan ketinggian air dan kecepatan angin yang terhitung. Algoritma subsistem sensor dapat dilihat pada diagram alir di bawah ini



Gambar 3.8. Diagram alir sub rutin sensor

3.3.3 Sub Rutin Pengiriman

Sub rutin pengiriman dirancang untuk mengirimkan pesan pada ponsel master yang berisi nilai dari level ketinggian air dan nilai kecepatan angin yang terhitung, pesan yang dikirimkan menunjukkan adanya perubahan level ketinggian air. Sub rutin pengiriman terdiri dari beberapa bagian yaitu :

a. Pengiriman perintah untuk mengirimkan SMS

Sebelum SMS dikirimkan, maka terlebih dahulu harus dikirimkan perintah AT Command untuk kirim SMS yaitu AT+CMGS=<jumlah oktet PDU>. <jumlah oktet PDU> diisi dengan jumlah pasangan dalam PDU yang terbentuk.

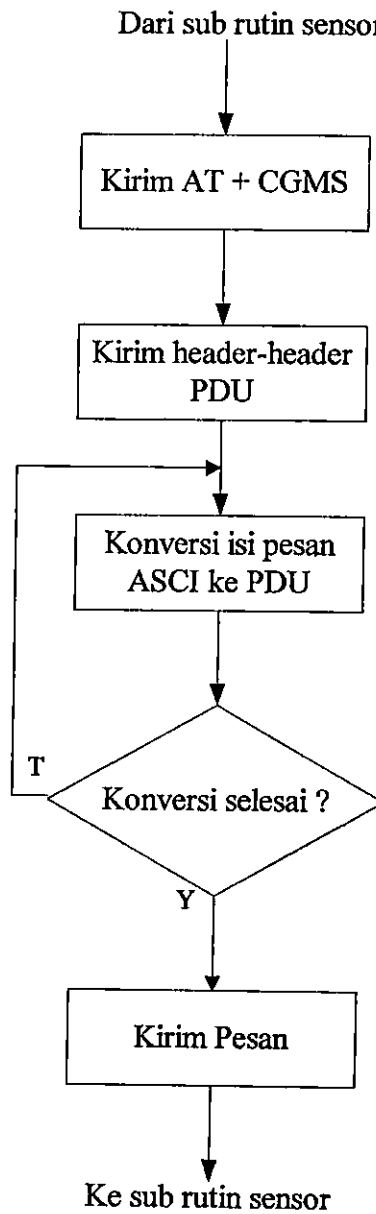
b. Pengiriman header-header PDU

Header-header PDU ini adalah kode-kode yang akan dikirimkan melalui PDU pengiriman tetapi tidak menyertakan isi dari pesan yang akan dikirimkan. Pada contoh diatas berarti header-header yang harus dikirimkan adalah 07 91 2658050000F0 11 00 0C 91 265836164900 00 00 FF 04. Sedangkan kode C830FB0D tidak disertakan karena kode ini adalah isi dari pesan dan akan dikirimkan. Kode akan dikirimkan pada bagian berikutnya.

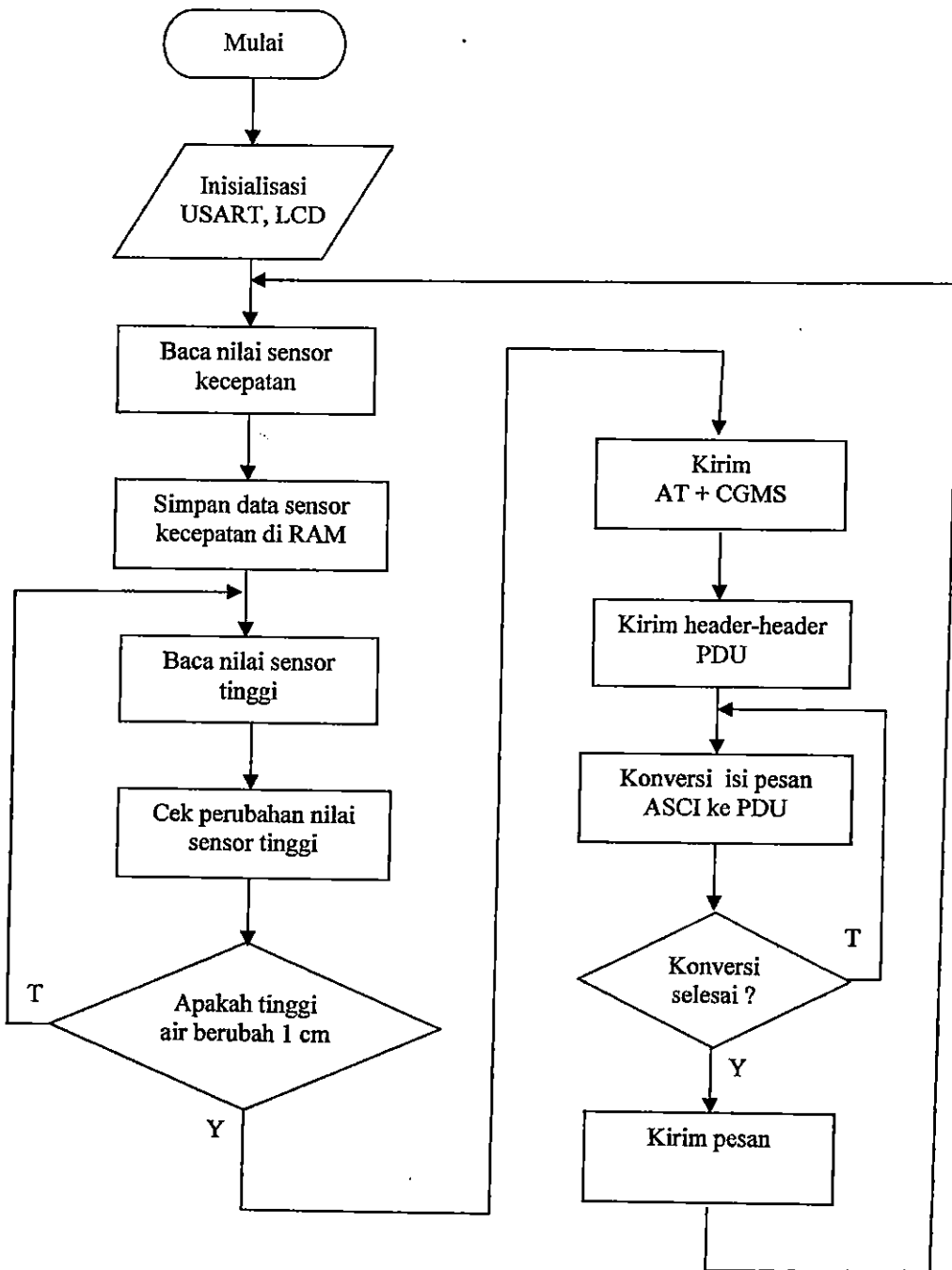
c. Konversi pesan dari format ASCII ke format PDU dan pengiriman pesan.

Pesan yang dikirimkan secara *default* akan memiliki format data ASCII 7 bit. Pada komunikasi dengan mikrokontroler maka data yang dikirimkan oleh ponsel harus dalam format PDU sehingga pesan yang berupa data ASCII harus diubah terlebih dahulu menjadi format PDU. Jika kita ingin mengirimkan pesan berupa kata 'Hallo' (format ASCII) maka isi pesan

harus diubah terlebih dahulu menjadi format PDU sehingga menjadi 'C830FB0D'. Pesan yang telah dikonversi akan dikirim satu-persatu.



Gambar 3.9 Diagram alir sub rutin pengiriman



Gambar 3.10 Diagram alir pendeteksi ketinggian air dan kecepatan angin laut

BAB IV

PENGUJIAN DAN HASIL

Pengujian sistem dilakukan untuk mengetahui apakah sistem yang dirancang berjalan sesuai dengan yang diinginkan. Pengujian dilaksanakan dalam 2 tahap yaitu pengujian perangkat keras dan pengujian secara keseluruhan.

4.1 Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan pada rangkaian sensor kecepatan. Rangkaian sensor kecepatan terdiri dari photo transistor dan dioda infra merah. Sensor ini berguna untuk mendeteksi kecepatan putaran baling-baling untuk mendapatkan kecepatan angin untuk satu kali pengukuran digunakan waktu 60 detik. Dalam pengujian ini dilakukan pengukuran secara manual, seperti kondisi transistor saat *cutt off* dan jenuh. Hasil pengukuran dapat dilihat pada tabel berikut.

Tabel 4.1. Pengukuran parameter rangkaian sensor kecepatan

Kondisi Basis Photo Transistor	Tegangan Pada Colector Emitor	Output Rangkaian Sensor
Tidak memperoleh sinar infra merah	4,9 Volt	0,1 Volt
Memperoleh sinar infra merah	0,4 Volt	4,9 Volt

Jika sinar infra merah terhalang menuju photo transistor maka tegangan outputnya 0,1 volt, sedangkan jika sinar infra merah tidak terhalang menuju photo transistor tegangan output sebesar 4,9 volt. Tegangan diberikan ke input IC *schmit*

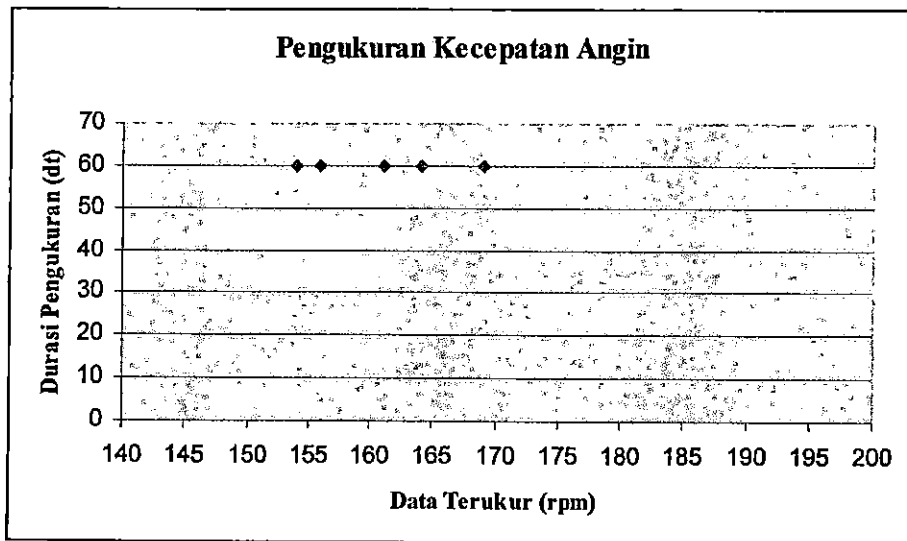
trigger inverter 74LS14. IC 74LS14 ini akan menginversikan tegangan 0,1 volt adalah logika '0' dan tegangan 4,9 volt adalah menjadi logika '1' pada keluarannya. Jadi pada saat sinar infra merah terhalang menuju photo transistor, maka pada pin P3.2 akan berlogika 0 (0 volt), sebaliknya jika sinar infra merah tidak terhalang menuju photo transistor, maka pin P3.2 akan berlogika 1 (5 volt).

Untuk mendapatkan kecepatan angin digunakan sebuah kipas angin untuk memutar baling-baling. Data yang dijadikan sebagai sampel adalah data yang dihasilkan dari putaran baling-baling dengan lama waktu pengukuran dalam pengujian 1 siklus selama 60 detik. Hasil pengujian dari pengukuran dapat dilihat pada tabel berikut.

Tabel 4.2. Pengujian kecepatan angin dalam 5 sample pengukuran.

No.	Durasi pengukuran (detik)	Data terukur (rpm)
1	60	169
2	60	156
3	60	154
4	60	164
5	60	161

Grafik yang menunjukkan titik-titik dari data yang didapatkan bisa dilihat pada gambar berikut ini. Dalam perancangan alat, untuk menghitung kecepatan dilakukan dengan mengumpulkan data kecepatan dalam waktu 60 detik, dimana selama 60 detik tersebut data tersebut disimpan oleh mikrokontroler.



Gambar 4.1. Grafik pengukurun kecepatan angin

Dari grafik tersebut dapat dilihat bahwa kecepatan angin tidak bersifat linear terhadap waktu pengukuran, hal ini dalam keadaan sebenarnya setiap menit kecepatan angin dapat berubah-ubah.

Sedangkan untuk rangkaian ultrasonik pengujian hanya dapat dilakukan dengan melakukan perhitungan. Rangkaian sensor ultrasonic yang digunakan adalah sensor ping yang berbentuk modul buatan Parallax. Modul sensor ping mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik (40 kHz) kemudian mendeteksi pantulannya. Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. Ping mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi Ping akan membuat *output low* pada pin SIG.

Untuk perhitungan sensor ultrasonik jarak sensor dengan dasar bak air ± 24 cm yang digunakan untuk menentukan jarak sensor ultrasonik dengan benda (air).

Hasil perhitungan sensor ultrasonik dengan ketinggian air sebagai jarak benda didapatkan waktu pantul dari sensor ultrasonik sesuai dengan teori.

$$\text{Jarak benda} = \text{Waktu pantul} \times 34,4$$

$$\text{Waktu pantul} = \text{Jarak benda} / 34,4$$

$$= 21 / 34,4$$

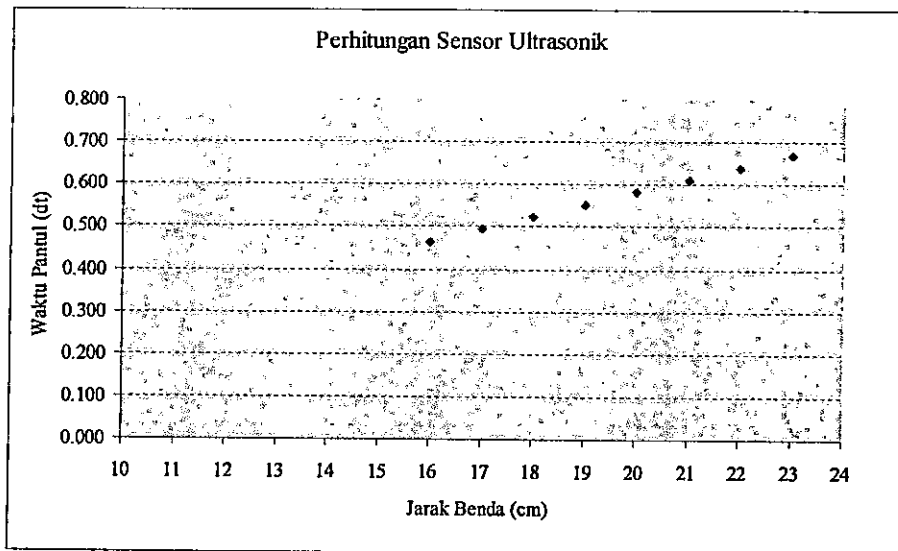
$$= 0.6104 \text{ detik}$$

Untuk hasil perhitungan sensor ultrasonik dengan jarak benda (air) dapat dilihat pada tabel dibawah ini .

Tabel 4.3. Perhitungan sensor ultrasonik

Tinggi Air (cm)	Jarak benda (cm)	Waktu pantul (detik)
1	23	0.6686
2	22	0.6395
3	21	0.6104
4	20	0.5813
5	19	0.5523
6	18	0.5232
7	17	0.4941
8	16	0.4651

Dari tabel 4.3 dapat diperoleh kesimpulan, bahwa lamanya waktu pantul untuk perubahan jarak benda 1 cm yang di deteksi oleh sensor ultrasonik diperlukan waktu sekitar 0.0291 detik.



Gambar 4.2. Grafik perbandingan jarak benda terhadap waktu pantul.

Berdasarkan grafik dari hasil perhitungan sensor ultrasonik, bahwa jarak benda (air) bersifat linear terhadap lamanya waktu pantul yang diterima pada sensor ultrasonik.

4.2 Pengujian Keseluruhan

Pengujian secara keseluruhan dilakukan dengan menggabungkan seluruh komponen yang menyusun sistem ini. Dalam pengujian secara keseluruhan berpatokan terhadap perubahan ketinggian air dalam bak air, sedangkan untuk kecepatan angin tetap dilakukan secara bersama-sama tetapi pada prinsipnya ketika kecepatan angin berubah dalam durasi 60 detik tidak akan mempengaruhi dalam proses pengiriman pesan SMS ke pada ponsel master. Dasar dari pengujian ini adalah mengkondisikan bak air saat terjadinya perubahan level ketinggian air dari ketinggian sebelumnya. Untuk pengujian ini dilakukan dalam beberapa kondisi yaitu pengujian pengiriman pesan, pengujian ketika level ketinggian air dalam bak naik dan pengujian ketika level ketinggian air dalam bak air turun.

4.2.1 Pengujian Pengiriman Pesan

Pengujian pengiriman pesan bertujuan untuk mengamati kemampuan mikrokontroler untuk mengirimkan pesan berdasarkan program yang telah dirancang. Pengujian dilakukan dalam 3 sample nomor posel yang berbeda. Hasil pengujian dapat dilihat pada tabel berikut ini

Tabel 4.4 Hasil pengujian pengiriman pesan dengan nomor pengirim berbeda

No	No. Pengirim	Pesan yang pada LCD	Pesan yang diterima	Status
1	085263192174	Tinggi = 01 cm	Tinggi = 01 cm, Kecepatan = 0163 rpm	Ok
		Kec. = 0163 rpm		
2	081380351751	Tinggi = 06 cm	Tinggi = 06 cm, Kecepatan = 0154 rpm	Ok
		Kec. = 0154 rpm		
3	085263741899	Tinggi = 08 cm	Tinggi = 08 cm, kecepatan = 0131 rpm	Ok
		Kec. = 0131 rpm		

Hasil pengujian menggunakan 3 nomor pengirim yang berbeda, dari tabel diatas pesan yang diterima oleh ponsel slave dapat menerima pesan yang sesuai dengan data yang ditampilkan pada LCD untuk 3 nomor pengirim yang berbeda. Berdasarkan pengujian yang telah dilakukan maka dapat disimpulkan bahwa mikrokontroler mampu melakukan pengiriman pesan dengan nomor pengirim berbeda kepada ponsel tujuan.

Untuk mengetahui lamanya waktu yang dibutuhkan untuk satu kali pengiriman pesan dengan nomor pengirim yang sama dan pengiriman pesan dengan nomor yang berbeda dalam waktu jam sibuk (jam 07.00 WIB – 17.00 WIB), dapat dilihat dalam tabel 4.5 dan tabel 4.6.

Tabel 4.5. Durasi pengiriman pesan dengan nomor pengirim yang sama

No	No. Pengirim	Pesan yang diterima	Durasi pengiriman (dt)
1	085263741899	Tinggi = 09 cm, Kecepatan = 0157 rpm	5.89
2	085263741899	Tinggi = 08 cm, Kecepatan = 0161 rpm	6.51
3	085263741899	Tinggi = 07 cm, Kecepatan = 0161 rpm	6.09
4	085263741899	Tinggi = 06 cm, Kecepatan = 0158 rpm	5.16
5	085263741899	Tinggi = 05 cm Kecepatan = 0154 rpm	6.43
Rata – rata pengiriman			6.02

Tabel 4.6. Durasi pengiriman pesan dengan nomor pengiriman berbeda

No	No. Pengirim	Pesan yang diterima	Durasi pengiriman (dk)
1	085263192174	Tinggi = 01 cm, Kecepatan = 0163 rpm	7.11
2	081380351751	Tinggi = 06 cm, Kecepatan = 0154 rpm	7.25
3	085263741899	Tinggi = 08 cm, kecepatan = 0131 rpm	6.43
Rata- rata pengiriman			6.69

Waktu yang dibutuhkan selama proses pengiriman pesan juga bervariasi. Keadaan ini tergantung dari trafik SMS pada saat itu.

4.2.2 Pengujian Level Ketinggian Air Naik

Ketinggian air dalam bak diset pada ketinggian 1 cm. Ketinggian air dapat dinaikan dengan menambahkan air ke dalam bak tersebut. Ketika waktu penambahan air dalam bak, apabila perubahan ketinggian air tersebut telah mencapai tinggi 2 cm dari permukaan bak, maka sensor akan mendeteksi adanya perubahan level ketinggian air dan dalam LCD akan menunjukkan tinggi air yang sebelumnya 1 cm menjadi 2 cm. Setelah sensor ultrasonik membaca adanya perubahan level ketinggian air maka akan terjadinya proses pengiriman SMS oleh mikrokontroler kepada ponsel master.

Pesan akan dikirimkan adalah data dari pembacaan sensor ultrasonik atau level ketinggian air yang terakhir dan juga data dari kecepatan angin terakhir yang terhitung selama 60 detik yang tersimpan dalam RAM mikrokontroler. Pesan tersebut berisikan kata "Tinggi = 002 cm, Kecepatan = 0165 rpm".

Pengujian ketika level ketinggian air naik dilakukan dalam 3 percobaan. Hasilnya adalah sebagai berikut

Tabel 4.7. Pengujian ketika level ketinggian air naik

No	Tinggi air sebelum (cm)	Tinggi air sesudah (cm)	Kecepatan angin sebelum (rpm)	Kecepatan angin sesudah (rpm)
1	1	2	165	165
2	2	3	156	164
3	3	4	164	164

Dari tabel diatas pada pengujian no.1 bahwa level ketinggian berubah dari 1 cm menjadi 2 cm, ini dikarenakan adanya proses penambahan volume air, sedangkan untuk kecepatan angin yang terhitung tetap dikarenakan lamanya waktu untuk perubahan ketinggian air dari 1 cm ke 2 cm hanya membutuhkan waktu sekitar 10 detik, sedangkan untuk perubahan kecepatan angin 1 siklus dibutuhkan waktu 60 detik. Untuk percobaan no.2, perubahan level ketinggian air tetap 1 cm sedangkan kecepatan angin berubah dari 156 rpm menjadi 164 rpm, hal ini berarti dalam perubahan level ketinggian air dari 2 cm ke tinggi 3 cm telah menggunakan waktu lebih dari 60 detik.

4.2.3 Pengujian Level Ketinggian Air Turun

Dalam pengujian ketika level ketinggian air turun ini dapat dilakukan dengan mengurangi volume air dalam bak melalui kran air. Dengan mengurangi volume air dalam bak melalui kran air akan menandakan adanya perubahan penurunan level ketinggian air dalam bak yang terbaca oleh sensor ultrasonik. Dengan adanya perubahan level ketinggian air yang terdeteksi oleh sensor ultrasonik dari level ketinggian yang sebelumnya, maka akan adanya proses pengiriman pesan oleh mikrokontroler ke ponsel master.

Pesan akan dikirimkan adalah data dari pembacaan sensor ultrasonik atau level ketinggian air yang terakhir dan juga data dari kecepatan angin terakhir yang tersimpan dalam RAM mikrokontroler. Pesan tersebut berisikan kata "Tinggi = 007 cm, Kecepatan = 0161 rpm":

Pengujian ketika level ketinggian air turun dilakukan dalam 3 percobaan. Hasilnya adalah sebagai berikut :

Tabel 4.8. Pengujian ketika level ketinggian air turun

No	Tinggi air sebelum (cm)	Tinggi air sesudah (cm)	Kecepatan angin sebelum (rpm)	Kecepatan angin sesudah (rpm)
1	8	7	161	161
2	7	6	161	159
3	6	5	159	159

Untuk hasil pengujian level ketinggian air turun dalam tabel 4.7 pada no.1 ketinggian level air dalam bak pada saat pengujian 8 cm, saat diturunkan dengan mengurangi volume air dalam bak menjadi tinggi air bak 7 cm, maka dalam hal ini akan terjadinya proses pengiriman pesan SMS dan kecepatan angin yang terukur baik sebelum dan sesudah perubahan level ketinggian air sebesar 161 rpm. Dalam percobaan yang kedua terlihat dalam tabel adanya perubahan kecepatan angin setelah adanya perubahan level ketinggian air pada saat tinggi air 7 cm berubah menjadi 6 cm, kecepatan angin yang sebelumnya terhitung 161 rpm dan sesudah perubahan level ketinggian air terhitung sebesar 159 rpm. Hal ini menandakan dalam perubahan level ketinggian air dari 7 cm menjadi 6 cm telah menggunakan waktu dalam perubahan level ketinggian air lebih dari 60 detik. Selanjutnya pada pengujian ketiga dari ketinggian level air dari 6 cm menjadi 5 cm, untuk kecepatan angin yang terhitung adalah tetap baik sebelum perubahan level ketinggian air maupun sesudah perubahan level ketinggian air.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perancangan sistem yang telah dilakukan maka dapat ditarik beberapa kesimpulan

1. SMS (*Short Message Service*) dapat digunakan sebagai sarana untuk pengiriman data dari hasil pengukuran level ketinggian air dan kecepatan angin laut secara *mobile*.
2. Kecepatan angin tidak bersifat linear, dikarenakan setiap saat kecepatan angin dapat berubah-ubah. Rata-rata kecepatan angin dalam durasi pengukuran 60 detik adalah 160 rpm
3. Rata-rata waktu yang diperlukan dalam pengiriman pesan SMS saat jam sibuk untuk nomor pengirim yang sama adalah 6.02 detik, dan untuk pengiriman peesan SMS dengan nomor berbeda adalah 6.69 detik.

5.2 Saran

Untuk mendapatkan hasil yang lebih baik dan lebih bermanfaat maka pada penelitian berikutnya disarankan sistem yang dirancang dapat dimanfaatkan dan diimplementasikan pada keadaan sebenarnya sebagai media alat pengukuran secara jarak jauh. Untuk merealisasikannya maka disarankan agar catu daya sistem dan ponsel diganti dengan sumber energi alternatif seperti tenaga air,

tenaga angin dan juga daerah pengukuran masih cakupan operator seluler sehingga selalu tersedia informasi data yang akan dimonitor dan diukur.

DAFTAR PUSTAKA

- [1] Istiyanto, Jazi Eko, Yeyen Efendy Rancangan Dan Implementasi Prototipe Sistem Kendali Jarak Jauh Berbasis AT89C52 Dan Layanan SMS GSM. Jurnal Ilmu Dasar, FMIPA Universitas Jember Vol. 5 No. 2, 2004 : 76-86
- [2] Istiyanto, Jazi Eko, Eko Purwadi. Alat Pemantau Suhu Jarak Jauh Berbasis SMS (An SMS-based Remote Temperature Monitoring Device). <http://www.alat%20PEMANTAU%20SUHU%20JARAK%20JAUH%20BERBASIS%20SMS.pdf> (diakses tanggal 4 oktober 2006)
- [3] Developer's Guide: SMS with the SMS PDU-mode. <http://www.gsmfavorites.com/documents/sms/packetformat/> (diakses tanggal 10 Februari 2007)
- [4] Sanjaya, Aryo. Mengirim SMS Dari PC. <http://www.bengkelprogram.com/data-artikel-175.last.bps> dipublikasikan pada 28 November 2005. (diakses tanggal 10 Februari 2007)
- [5] Kamil, Husnil. 2005. Tugas Akhir Penggunaan Short Message Service (SMS) sebagai Media Transfer Perintah Operasi Motor DC. Padang. Universitas Andalas.
- [6] Note Application : Pengukur jarak dengan gelombang ultrasonik.
- [7] Wardhana, Lingga. 2006. Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware dan Aplikasi. Penerbit Andi ; Yogyakarta.
- [8] Malik, Moh. Ibnu, Anistardi, 1997, "*Bereksperimen dengan Mikrokontroler 8031*", PT. Elex Media Komputindo, Jakarta, 1997.

LAMPIRAN 1
LISTING PROGRAM

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

```

DISPCLR      EQU 00000001B
FUNCSET      EQU 00111000B
ENTRMOD      EQU 00000110B
DISPON       EQU 00001100B

```

```

DSEG

```

```

ORG 0030H

```

```

vtinggi:    ds 1
tinggi:     ds 1
VRPM_Digit1: ds 1
VRPM_Digit2: ds 1
VRPM_Digit3: ds 1
VRPM_Digit4: ds 1
RPM_Digit1: ds 1
RPM_Digit2: ds 1
RPM_Digit3: ds 1
RPM_Digit4: ds 1

```

```

CSEG

```

```

ORG 0000H

```

```

LJMP 0100h

```

```

org 0003h

```

```

LJMP INT0_ISR

```

```

org 0013h

```

```

LJMP INT1_ISR

```

```

INT0_ISR:

```

```

    jb  p3.2,akhir1
    acall delay_20ms
    jnb p3.2,$
    acall delay_20ms
    acall tambah_RPM
    acall tampilan_RPM
    acall ultrasonik_ukur_tinggi

```

```

akhir1:    reti

```

```

tambah_RPM:

```

```

    mov a,VRPM_Digit1
    inc a
    cjne a,#10,lanjut1
    mov a,#0
    push acc
    mov a,VRPM_Digit2
    inc a
    mov VRPM_Digit2,a
    pop acc

```

```

lanjut1:  mov VRPM_Digit1,a

```

```

    mov a,VRPM_Digit2
    cjne a,#10,lanjut2
    mov a,#0
    push acc
    mov a,VRPM_Digit3
    inc a
    mov VRPM_Digit3,a
    pop acc
lanjut2: mov VRPM_Digit2,a
    mov a,VRPM_Digit3
    cjne a,#10,lanjut3
    mov a,#0
    push acc
    mov a,VRPM_Digit4
    inc a
    mov VRPM_Digit4,a
    pop acc
lanjut3: mov VRPM_Digit3,a
    mov a,VRPM_Digit4
    cjne a,#10,lanjut4
    mov a,#0
lanjut4: mov VRPM_Digit4,a
    ret

    mov a,VRPM_Digit1
    mov RPM_Digit1,a
    acall delay_10ms
    mov VRPM_Digit1,#0

    mov a,VRPM_Digit2
    acall delay_10ms
    mov RPM_Digit2,a
    mov VRPM_Digit2,#0

    acall delay_10ms
    mov a,VRPM_Digit3
    mov RPM_Digit3,a
    acall delay_10ms
    mov VRPM_Digit3,#0

    mov a,VRPM_Digit4
    acall delay_10ms
    mov RPM_Digit4,a
    mov VRPM_Digit4,#0
    acall delay_10ms

```

```

    acall tampilan_RPM
    ret
    org 0100h
main: mov SP,#50h
    acall delay_10ms
    LCALL Init_LCD
    MOV DPTR,#COVER1A
    lcall PRINTSTRING1
    MOV DPTR,#COVER1B
    lcall PRINTSTRING2
    lcall delay_1dtk
    MOV DPTR,#COVER2A
    lcall PRINTSTRING1
    MOV DPTR,#COVER2B
    lcall PRINTSTRING2
    lcall delay_1dtk
    MOV DPTR,#COVER3A
    lcall PRINTSTRING1
    MOV DPTR,#COVER3B
    lcall PRINTSTRING2
    lcall delay_1dtk
    MOV DPTR,#COVER4A
    lcall PRINTSTRING1
    MOV DPTR,#COVER4B
    lcall PRINTSTRING2
    lcall delay_1dtk
    MOV DPTR,#COVER5A
    lcall PRINTSTRING1
    MOV DPTR,#COVER5B
    lcall PRINTSTRING2
    lcall delay_1dtk
    MOV DPTR,#MENU1A
    lcall PRINTSTRING1
    MOV DPTR,#MENU1B
    lcall PRINTSTRING2
    mov VRPM_Digit1,#0
    mov VRPM_Digit2,#0
    mov VRPM_Digit3,#0
    mov VRPM_Digit4,#0
    mov RPM_Digit1,#0
    mov RPM_Digit2,#0
    mov RPM_Digit3,#0
    mov RPM_Digit4,#0
    mov vtinggi,#1
    mov tinggi,#1

```

```

lcall delay_1dtk
clr p1.0
MOV A,#00h
MOV tmod,#21h
MOV th1,#0fdh
MOV scon,#50h
MOV pcon,#80h
SETB tr1
setb ex0
setb ex1
setb ea
mov r5,#59
mulai: acall delay_500ms
acall ultrasonik_ukur_tinggi
acall delay_500ms
djnz r5,mulai
mov r5,#59
ajmp mulai

```

ULTRASONIK_UKUR_TINGGI:

```

clr p1.0
acall delay_20ms
mov tmod,#21h
clr tr0
mov tl0,#0
mov th0,#0
setb p1.0
nop
nop
nop
nop
nop
nop
clr p1.0
acall delay_750
setb p1.0
setb tr0
nop
nop
nop
nop
nop
cek_p10: jb p1.0,cek_p10
clr tr0
MOV A,TH0
MOV B,#5

```

```

MUL AB
MOV R4,A
MOV B,#51
MOV A,TL0
DIV AB
ADD A,R4
MOV vtinggi,A
mov a,#22
subb a,vtinggi
jc LCD_OOR
mov vtinggi,a
push acc
xrl a,#0ffh
mov p0,a
pop acc
acall delay_10ms
acall tampilan_tinggi
acall delay_10ms
mov a,vtinggi
cjne a,tinggi,kirim_sms
acall delay_10ms

```

akhir1A: ret

LCD_OOR:

```

MOV A,#8BH ; PESAN TEMPAT DI KOLOM 12 BARIS 1
ACALL CONTROLOUT
MOV A,#' '
ACALL DATAOUT
MOV A,#8AH ; PESAN TEMPAT DI KOLOM 11 BARIS 1
ACALL CONTROLOUT
MOV A,#' '
ACALL DATAOUT
MOV A,#89H ; PESAN TEMPAT DI KOLOM 10 BARIS 1
ACALL CONTROLOUT
MOV A,#' '
ACALL DATAOUT
ret

```

KIRIM_SMS:

```

MOV tmod,#21h
MOV th1,#0fdh
MOV scon,#50h
MOV pcon,#80h
SETB tr1
acall delay_100ms
MOV TINGGI,A

```



```

AT_TINGGI:
    MOV  DPTR,#cmd_tinggi
    MOV  R2,#11
    lcall kirim1
    lcall delay_100ms
    MOV  DPTR,#sms_data_tinggi
    MOV  R2,#58
    lcall kirim1
    mov  a,vtinggi
tinggi0:  cjne a,#0,tinggi1
    MOV  DPTR,#data_tinggi0
    ajmp kirim_tinggi
tinggi1:  cjne a,#1,tinggi2
    MOV  DPTR,#data_tinggi1
    ajmp kirim_tinggi
tinggi2:  cjne a,#2,tinggi3
    MOV  DPTR,#data_tinggi2
    ajmp kirim_tinggi
tinggi3:  cjne a,#3,tinggi4
    MOV  DPTR,#data_tinggi3
    ajmp kirim_tinggi
tinggi4:  cjne a,#4,tinggi5
    MOV  DPTR,#data_tinggi4
    ajmp kirim_tinggi
tinggi5:  cjne a,#5,tinggi6
    MOV  DPTR,#data_tinggi5
    ajmp kirim_tinggi
tinggi6:  cjne a,#6,tinggi7
    MOV  DPTR,#data_tinggi6
    ajmp kirim_tinggi
tinggi7:  cjne a,#7,tinggi8
    MOV  DPTR,#data_tinggi7
    ajmp kirim_tinggi
tinggi8:  cjne a,#8,tinggi9
    MOV  DPTR,#data_tinggi8
    ajmp kirim_tinggi
tinggi9:  cjne a,#9,tinggi10
    MOV  DPTR,#data_tinggi9
    ajmp kirim_tinggi
tinggi10: cjne a,#10,tinggi11
    MOV  DPTR,#data_tinggi10
    ajmp kirim_tinggi
tinggi11: cjne a,#11,tinggi12
    MOV  DPTR,#data_tinggi11
    ajmp kirim_tinggi

```

```

tinggi12: cjne a,#12,tinggi13
          MOV  DPTR,#data_tinggi12
          ajmp kirim_tinggi
tinggi13: cjne a,#13,tinggi14
          MOV  DPTR,#data_tinggi13
          ajmp kirim_tinggi
tinggi14: cjne a,#14,tinggi15
          MOV  DPTR,#data_tinggi14
          ajmp kirim_tinggi
tinggi15: cjne a,#15,tinggi16
          MOV  DPTR,#data_tinggi15
          ajmp kirim_tinggi
tinggi16: cjne a,#16,tinggi17
          MOV  DPTR,#data_tinggi16
          ajmp kirim_tinggi
tinggi17: cjne a,#17,tinggi18
          MOV  DPTR,#data_tinggi17
          ajmp kirim_tinggi
tinggi18: cjne a,#18,tinggi19
          MOV  DPTR,#data_tinggi18
          ajmp kirim_tinggi
tinggi19: cjne a,#19,tinggi20
          MOV  DPTR,#data_tinggi19
          ajmp kirim_tinggi
tinggi20: cjne a,#20,tinggi21
          MOV  DPTR,#data_tinggi20
          ajmp kirim_tinggi
tinggi21: cjne a,#21,tinggi22
          MOV  DPTR,#data_tinggi21
          ajmp kirim_tinggi
tinggi22: cjne a,#22,tinggi22a
tinggi22a:MOV  DPTR,#data_tinggi22

kirim_tinggi: MOV  R2,#4
              lcall kirim1
              lcall delay_100ms
              MOV  DPTR,#sms_data_RPM1
              MOV  R2,#28
              lcall kirim1

;CEK DIGIT4
          mov  a,RPM_Digit4
C_RPM_Digit4_0: cjne a,#0,C_RPM_Digit4_1
              MOV  DPTR,#RPM_Digit4_0
              ajmp kirim_digit4

```

```

C_RPM_Digit4_1: cjne a,#1,C_RPM_Digit4_2
                 MOV  DPTR,#RPM_Digit4_1
                 ajmp kirim_digit4
C_RPM_Digit4_2: cjne a,#2,C_RPM_Digit4_2a
C_RPM_Digit4_2a: MOV  DPTR,#RPM_Digit4_2
Kirim_Digit4:   mov  r2,#2
                 lcall kirim1
                 lcall delay_100ms

;CEK DIGIT3
                 mov  a,RPM_Digit3
C_RPM_Digit3_0: cjne a,#0,C_RPM_Digit3_1
                 MOV  DPTR,#RPM_Digit3_0
                 ajmp kirim_digit3
C_RPM_Digit3_1: cjne a,#1,C_RPM_Digit3_2
                 MOV  DPTR,#RPM_Digit3_1
                 ajmp kirim_digit3
C_RPM_Digit3_2: cjne a,#2,C_RPM_Digit3_3
                 MOV  DPTR,#RPM_Digit3_2
                 ajmp kirim_digit3
C_RPM_Digit3_3: cjne a,#3,C_RPM_Digit3_4
                 MOV  DPTR,#RPM_Digit3_3
                 ajmp kirim_digit3
C_RPM_Digit3_4: cjne a,#4,C_RPM_Digit3_5
                 MOV  DPTR,#RPM_Digit3_4
                 ajmp kirim_digit3
C_RPM_Digit3_5: cjne a,#5,C_RPM_Digit3_6
                 MOV  DPTR,#RPM_Digit3_5
                 ajmp kirim_digit3
C_RPM_Digit3_6: cjne a,#6,C_RPM_Digit3_7
                 MOV  DPTR,#RPM_Digit3_6
                 ajmp kirim_digit3
C_RPM_Digit3_7: cjne a,#7,C_RPM_Digit3_8
                 MOV  DPTR,#RPM_Digit3_7
                 ajmp kirim_digit3
C_RPM_Digit3_8: cjne a,#8,C_RPM_Digit3_9
                 MOV  DPTR,#RPM_Digit3_8
                 ajmp kirim_digit3
C_RPM_Digit3_9: cjne a,#9,C_RPM_Digit3_9a
C_RPM_Digit3_9a: MOV  DPTR,#RPM_Digit3_9
Kirim_Digit3:   mov  r2,#2
                 lcall kirim1
                 lcall delay_100ms

;CEK DIGIT2

```

```

        mov a,RPM_Digit2
C_RPM_Digit2_0: cjne a,#0,C_RPM_Digit2_1
                MOV DPTR,#RPM_Digit2_0
                ajmp kirim_digit2
C_RPM_Digit2_1: cjne a,#1,C_RPM_Digit2_2
                MOV DPTR,#RPM_Digit2_1
                ajmp kirim_digit2
C_RPM_Digit2_2: cjne a,#2,C_RPM_Digit2_3
                MOV DPTR,#RPM_Digit2_2
                ajmp kirim_digit2
C_RPM_Digit2_3: cjne a,#3,C_RPM_Digit2_4
                MOV DPTR,#RPM_Digit2_3
                ajmp kirim_digit2
C_RPM_Digit2_4: cjne a,#4,C_RPM_Digit2_5
                MOV DPTR,#RPM_Digit2_4
                ajmp kirim_digit2
C_RPM_Digit2_5: cjne a,#5,C_RPM_Digit2_6
                MOV DPTR,#RPM_Digit2_5
                ajmp kirim_digit2
C_RPM_Digit2_6: cjne a,#6,C_RPM_Digit2_7
                MOV DPTR,#RPM_Digit2_6
                ajmp kirim_digit2
C_RPM_Digit2_7: cjne a,#7,C_RPM_Digit2_8
                MOV DPTR,#RPM_Digit2_7
                ajmp kirim_digit2
C_RPM_Digit2_8: cjne a,#8,C_RPM_Digit2_9
                MOV DPTR,#RPM_Digit2_8
                ajmp kirim_digit2
C_RPM_Digit2_9: cjne a,#9,C_RPM_Digit2_9a
C_RPM_Digit2_9a: MOV DPTR,#RPM_Digit2_9
Kirim_Digit2:  mov r2,#2
                lcall kirim1
                lcall delay_100ms

```

```

;CEK DIGIT1

```

```

        mov a,RPM_Digit1
C_RPM_Digit1_0: cjne a,#0,C_RPM_Digit1_1
                MOV DPTR,#RPM_Digit1_0
                ajmp kirim_digit1
C_RPM_Digit1_1: cjne a,#1,C_RPM_Digit1_2
                MOV DPTR,#RPM_Digit1_1
                ajmp kirim_digit1
C_RPM_Digit1_2: cjne a,#2,C_RPM_Digit1_3
                MOV DPTR,#RPM_Digit1_2
                ajmp kirim_digit1

```

```

C_RPM_Digit1_3: cjne a,#3,C_RPM_Digit1_4
MOV DPTR,#RPM_Digit1_3
ajmp kirim_digit1
C_RPM_Digit1_4: cjne a,#4,C_RPM_Digit1_5
MOV DPTR,#RPM_Digit1_4
ajmp kirim_digit1
C_RPM_Digit1_5: cjne a,#5,C_RPM_Digit1_6
MOV DPTR,#RPM_Digit1_5
ajmp kirim_digit1
C_RPM_Digit1_6: cjne a,#6,C_RPM_Digit1_7
MOV DPTR,#RPM_Digit1_6
ajmp kirim_digit1
C_RPM_Digit1_7: cjne a,#7,C_RPM_Digit1_8
MOV DPTR,#RPM_Digit1_7
ajmp kirim_digit1
C_RPM_Digit1_8: cjne a,#8,C_RPM_Digit1_9
MOV DPTR,#RPM_Digit1_8
ajmp kirim_digit1
C_RPM_Digit1_9: cjne a,#9,C_RPM_Digit1_9a
C_RPM_Digit1_9a: MOV DPTR,#RPM_Digit1_9
Kirim_Digit1: mov r2,#2
lcall kirim1
;Kirim data sms terakhir
lcall delay_100ms
MOV DPTR,#sms_data_RPM2
MOV R2,#9
lcall kirim1
ret

```

```

cmd_tinggi: DB 'AT+CMGS=46',13
sms_data_tinggi: DB
'0691261801000011000D91265862132187F50000AA23D4B4FB7C4EF740'
data_tinggi0: DB '3018'
data_tinggi1: DB 'B018'
data_tinggi2: DB '3019'
data_tinggi3: DB 'B019'
data_tinggi4: DB '301A'
data_tinggi5: DB 'B01A'
data_tinggi6: DB '301B'
data_tinggi7: DB 'B01B'
data_tinggi8: DB '301C'
data_tinggi9: DB 'B01C'
data_tinggi10: DB '3118'
data_tinggi11: DB 'B118'
data_tinggi12: DB '3119'

```

data_tinggi13: DB 'B119'
data_tinggi14: DB '311A'
data_tinggi15: DB 'B11A'
data_tinggi16: DB '311B'
data_tinggi17: DB 'B11B'
data_tinggi18: DB '311C'
data_tinggi19: DB 'B11C'
data_tinggi20: DB '3218'
data_tinggi21: DB 'B218'
data_tinggi22: DB '3219'
sms_data_RPM1: DB '68DC668196E571191EA687DDA01E'

RPM_digit4_0: DB '08'
RPM_digit4_1: DB '28'
RPM_digit4_2: DB '48'
RPM_digit3_0: DB '06'
RPM_digit3_1: DB '16'
RPM_digit3_2: DB '26'
RPM_digit3_3: DB '36'
RPM_digit3_4: DB '46'
RPM_digit3_5: DB '56'
RPM_digit3_6: DB '66'
RPM_digit3_7: DB '76'
RPM_digit3_8: DB '86'
RPM_digit3_9: DB '96'
RPM_digit2_0: DB '83'
RPM_digit2_1: DB '8B'
RPM_digit2_2: DB '93'
RPM_digit2_3: DB '9B'
RPM_digit2_4: DB 'A3'
RPM_digit2_5: DB 'AB'
RPM_digit2_6: DB 'B3'
RPM_digit2_7: DB 'BB'
RPM_digit2_8: DB 'C3'
RPM_digit2_9: DB 'CB'
RPM_digit1_0: DB 'C1'
RPM_digit1_1: DB 'C5'
RPM_digit1_2: DB 'C9'
RPM_digit1_3: DB 'CD'
RPM_digit1_4: DB 'D1'
RPM_digit1_5: DB 'D5'
RPM_digit1_6: DB 'D9'
RPM_digit1_7: DB 'DD'
RPM_digit1_8: DB 'E1'
RPM_digit1_9: DB 'E5'

sms_data_RPM2: DB '4072781B',26

KIRIM1:

```
CLR A
MOVC A,@A+dptr
MOV sbuf,A
JNB TI,$
CLR TI
INC DPTR
DJNZ R2,kirim1
ret
```

TAMPILAN_TINGGI:

```
MOV B,#10
DIV AB
MOV R2,B ; R2 : SATUAN
MOV B,#10
DIV AB
MOV R1,B ; R1 : PULUHAN
MOV R0,A ; R0 : RATUSAN
MOV A,#8BH ; PESAN TEMPAT DI KOLOM 12 BARIS 1
ACALL CONTROLOUT
MOV A,R2
ADD A,#30H ; ASCII DARI SATUAN
ACALL DATAOUT
MOV A,#8AH ; PESAN TEMPAT DI KOLOM 11 BARIS 1
ACALL CONTROLOUT
MOV A,R1
ADD A,#30H ; ASCII DARI PULUHAN
ACALL DATAOUT
MOV A,#89H ; PESAN TEMPAT DI KOLOM 10 BARIS 1
ACALL CONTROLOUT
MOV A,R0
ADD A,#30H ; ASCII DARI RATUSAN
ACALL DATAOUT
RET
```

Tampilan_RPM:

```
MOV A,#0CBH ; PESAN TEMPAT DI KOLOM 11 BARIS 1
ACALL CONTROLOUT
MOV A,VRPM_Digit1
ADD A,#30H ; ASCII Digit1
ACALL DATAOUT
MOV A,#0CAH ; PESAN TEMPAT DI KOLOM 10 BARIS 1
ACALL CONTROLOUT
```

```

MOV  A,VRPM_Digit2
ADD  A,#30H      ; ASCII Digit2
ACALL DATAOUT
MOV  A,#0C9H    ; PESAN TEMPAT DI KOLOM 9 BARIS 1
ACALL CONTROLOUT
MOV  A,VRPM_Digit3
ADD  A,#30H      ; ASCII Digit3
ACALL DATAOUT
MOV  A,#0C8H    ; PESAN TEMPAT DI KOLOM 8 BARIS 1
ACALL CONTROLOUT
MOV  A,VRPM_Digit4
ADD  A,#30H      ; ASCII Digit4
ACALL DATAOUT
RET

```

```

;-----
;  PROSEDUR PEMESANAN TEMPAT
;-----

```

```

;RS : 0      : P3.6,   E : 0 - 1 - 0   : P3.7,   DATA      : P2

```

```
CONTROLOUT:
```

```

CLR  P3.6      ; RS = 0
ACALL DELAY
CLR  P3.7      ; E = 0
ACALL DELAY
SETB P3.7      ; E = 1
ACALL DELAY
MOV  P2,A
ACALL DELAY
CLR  P3.7      ; E = 0
ACALL DELAY
RET

```

```

;-----
;  PROSEDUR PENEMPATAN DATA
;-----

```

```

;RS : 0      : P3.6,   E : 0 - 1 - 0   : P3.7,   DATA      : P2

```

```
DATAOUT:
```

```

SETB P3.6      ; RS = 1
ACALL DELAY
CLR  P3.7      ; E = 0
ACALL DELAY
SETB P3.7      ; E = 1
ACALL DELAY
MOV  P2,A
ACALL DELAY
CLR  P3.7      ; E = 0
ACALL DELAY

```


RET

INIT_LCD:

```
MOV A,#FUNCSET
ACALL CONTROLOUT
ACALL DELAY_INIT_LCD
MOV A,#DISPON
ACALL CONTROLOUT
ACALL DELAY_INIT_LCD
MOV A,#DISPCLR
ACALL CONTROLOUT
ACALL DELAY_INIT_LCD
MOV A,#ENTRMOD
ACALL CONTROLOUT
ACALL DELAY_INIT_LCD
RET
```

;-----
; PROCEDURE LCD BARIS KOLOM LCD
;-----

POSISI2_1:

```
MOV A,#1
```

POSISI2:

```
ADD A,#11000000B ; BARIS KEDUA
DEC A
ACALL CONTROLOUT
ACALL DELAY_INIT_LCD
RET
```

POSISI1_1:

```
MOV A,#1
```

POSISI1:

```
ADD A,#10000000B ; BARIS PERTAMA
DEC A
ACALL CONTROLOUT
ACALL DELAY_INIT_LCD
RET
```

PRINTSTRING2:

```
ACALL POSISI2_1
CLR A
MOVC A,@A+DPTR
JNZ PRINTSTRINGLOOP
```

PRINTSTRINGLOOP:

```
ACALL DATAOUT
ACALL DELAY_INIT_LCD
```

```

INC DPTR
CLR A
MOVC A,@A+DPTR
JNZ PRINTSTRINGLOOP
RET

```

```

PRINTSTRING1:
ACALL POSISI1_1
CLR A
MOVC A,@A+DPTR
JNZ PRINTSTRINGLOOP2

```

```

PRINTSTRINGLOOP2:
LCALL DATAOUT
ACALL DELAY_INIT_LCD
INC DPTR
CLR A
MOVC A,@A+DPTR
JNZ PRINTSTRINGLOOP2
RET

```

```

;-----
; PROSEDUR DELAY, REGISTER = R7
;-----

```

```

delay_1dtk:
    mov    r7,#0
ulangi1: acall delay_10ms
    cjne  r7,#100,ulangi1
    ret

```

```

delay_2dtk:
    mov    r7,#0
ulangi2: acall delay_10ms
    cjne  r7,#200,ulangi2
    ret

```

```

delay_20ms:
    mov    r7,#0
ulangi3: acall delay_10ms
    cjne  r7,#2,ulangi3
    ret

```

```

delay_100ms:
    mov    r7,#0
ulangi4: acall delay_10ms
    cjne  r7,#10,ulangi4
    ret

```

```

delay_200ms:
    mov    r7,#0
ulangi5:    acall delay_10ms
            cjne  r7,#20,ulangi5
            ret

```

```

delay_500ms:
    mov    r7,#0
ulangi6:    acall delay_10ms
            cjne  r7,#50,ulangi6
            ret

```

```

delay_10ms:
    mov    th2,#0d8h
    mov    tl2,#0efh
    setb  tr2
ulang20:    nop
            jbc  tf2,hitung20
            sjmp ulang20
hitung20:   inc  r7
            ret

```

```

delay_750:  mov  tmod,#21h
            mov  th2,#0fdh ; fc
            mov  tl2,#11h ; 11H
            setb tr2
ulang_750:  nop
            jbc  tf2,hitung_750
            sjmp ulang_750
hitung_750: ret

```

```

;-----
;   PROSEDUR DELAY LCD
;-----

```

```

DELAY_INIT_LCD: ; delay 512 x
                mov  tmod,#21h
                mov  th2,#0fdh
                mov  tl2,#0ffh
                setb  tr2
ulang40:        nop
                jbc  tf2,hitung40
                sjmp ulang40
hitung40:       inc  r7
                ret

```

```
;-----  
; PROCEDURE DELAY  
;-----
```

```
DELAY: ;delay 125 x  
      MOV R7,#125  
      DJNZ R7,$  
      RET
```

```
;0123456789ABCDEF
```

```
COVER1A: DB 'RANCANGAN PROTO-',0  
COVER1B: DB 'TYPE PENDETEKSI',0  
COVER2A: DB ' KETINGGIAN AIR ',0  
COVER2B: DB ' DAN KECEPATAN ',0  
COVER3A: DB ' ANGIN LAUT DGN ',0  
COVER3B: DB ' LAYANAN SMS ',0  
COVER4A: DB 'Oleh : Benny W.F',0  
COVER4B: DB ' BP: 06 975 006 ',0  
COVER5A: DB ' ELEKTRO UNAND ',0  
COVER5B: DB ' 2008 ',0  
MENU1A: DB 'Tinggi: 000 cm ',0  
MENU1B: DB 'Kec. : 0000 rpm',0  
      END
```

LAMPIRAN 2

DATASHEET MIKROKONTROLER

AT89C52

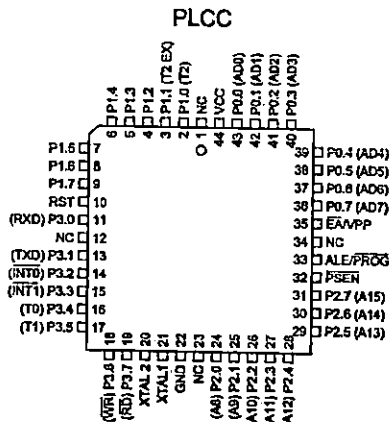
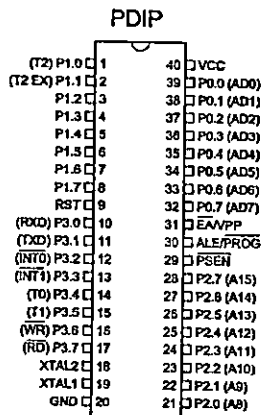
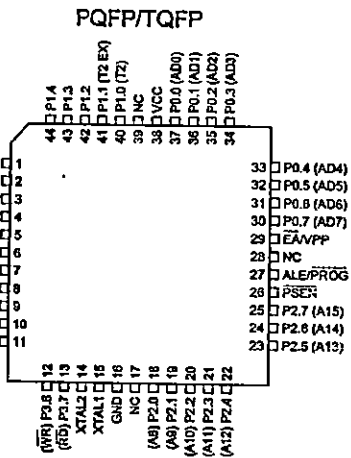
Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 6 x 8-bit Internal RAM
- Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K Bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout. On-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU and Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer that provides a highly-flexible and cost-effective solution to many embedded control applications.

Configurations



8-bit Microcontroller with 8K Bytes Flash

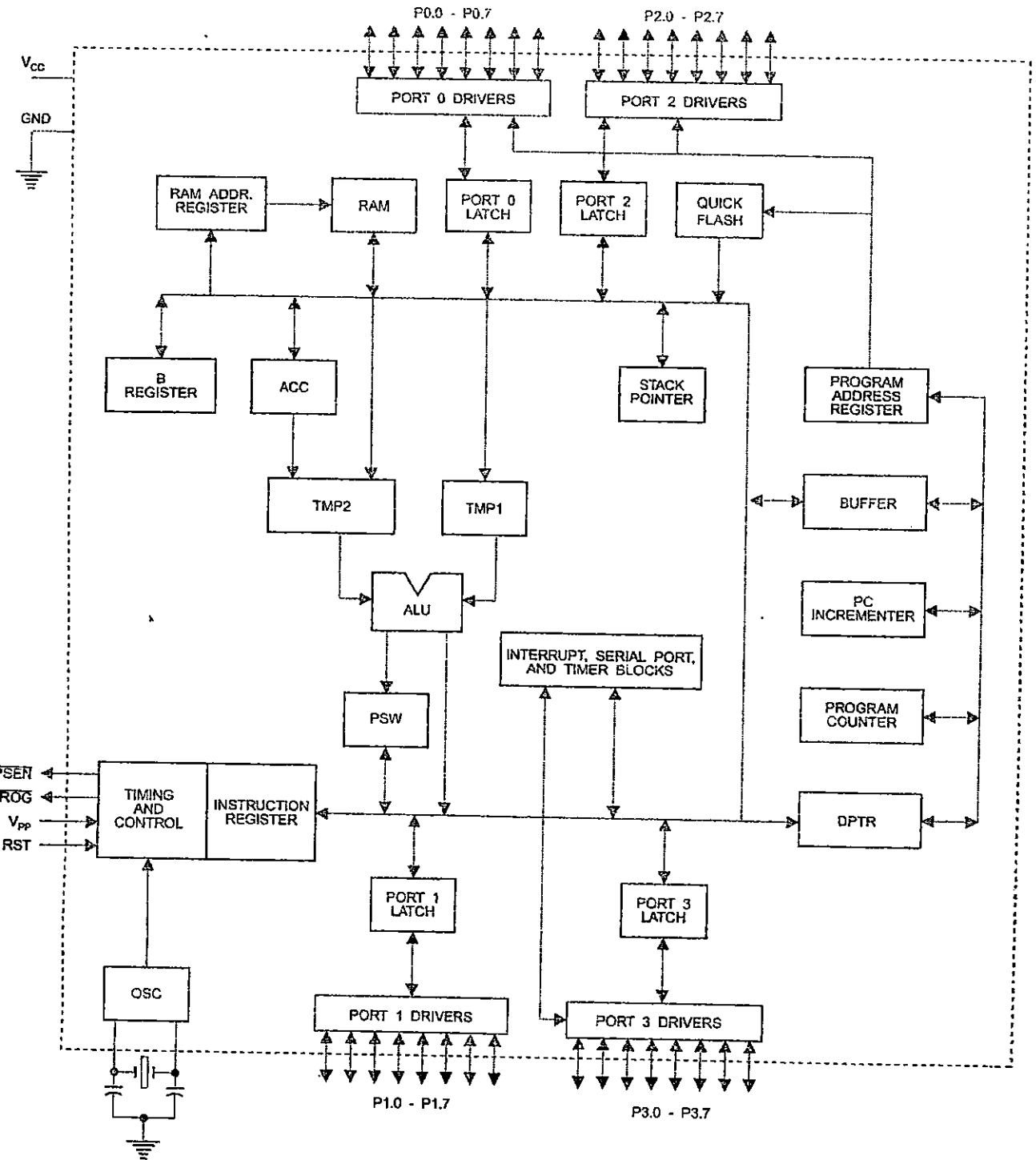
AT89C52

Not Recommended
for New Designs.
Use AT89S52.

Rev. 03132-02/00



Block Diagram



AT89C52

Special Function Registers

Map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Programmer software should not write 1s to these unlisted locations, since they may be used in future products to invoke

new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H		Reset Value = 0000 0000B						
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T $\bar{2}$	CP/RL $\bar{2}$
	7	6	5	4	3	2	1	0
Symbol	Function							
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.							
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).							
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.							
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.							
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.							
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.							
C/T $\bar{2}$	Timer or counter select for Timer 2. C/T $\bar{2}$ = 0 for timer function. C/T $\bar{2}$ = 1 for external event counter (falling edge triggered).							
CP/RL $\bar{2}$	Capture/Reload select. CP/RL $\bar{2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL $\bar{2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.							

RAM Memory

The AT89C52 implements 256 bytes of on-chip RAM. The lower 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction

specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```



LAMPIRAN 3

DATASHEET SENSOR PING

DT-51

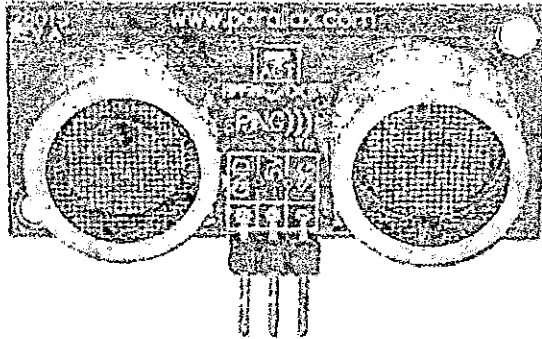
DT-51 *Application Note* AN73 – Pengukur Jarak dengan Gelombang Ultrasonik Oleh: Tim IE

Aplikasi ini membahas perencanaan dan pembuatan alat untuk mengukur jarak sebuah benda solid dengan cukup presisi dan tanpa kontak fisik. Alat ini direncanakan dan dibuat agar dapat digunakan secara *portable* ataupun dengan terhubung ke komputer melalui COM port. Aplikasi ini menggunakan modul sensor PING)))™ Ultrasonic Range Finder yang dapat dihubungkan dengan mudah ke DT-51™ Low Cost Micro System / Low Cost Nano System melalui 1 pin I/O saja. Bahasa yang digunakan adalah BASIC dengan bantuan BASCOM-8051[®].

Berdasarkan *datasheet*, modul PING)))™ ini dapat digunakan untuk mengukur jarak benda sejauh 3 cm sampai 300 cm. Sistem ini dapat diterapkan pada robot sebagai pengindra adanya halangan.

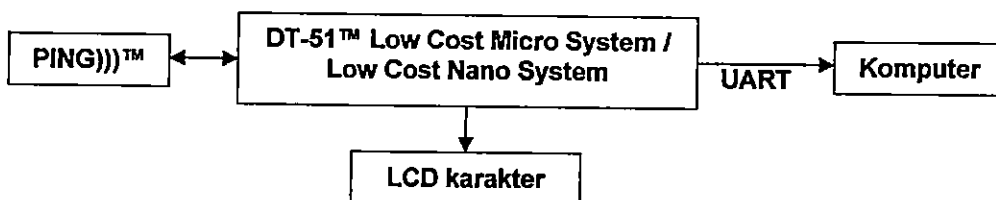
Komponen yang diperlukan:

- 1 bh PING)))™ Ultrasonic Range Finder
- 1 bh DT-51™ Low Cost Micro System / Low Cost Nano System
- 1 bh LCD karakter 24x2 tanpa backlight (yang kompatibel dengan driver HD44780)



Gambar 1
Sensor PING)))™

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



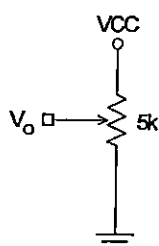
Gambar 2
Blok Diagram AN73

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-51™ Low Cost Micro System / Low Cost Nano System	LCD Karakter
GND	GND
V _{cc}	V _{cc}
-	V _o *
P1.0	RS
GND	R/W
P1.2	E
P1.4	DB4
P1.5	DB5
P1.6	DB6
P1.7	DB7

Tabel 1

Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan LCD Karakter



Gambar 3

Hubungan Pin V_o LCD ke VR 5k ohm

Sensor PING)))™	DT-51™ Low Cost Micro System / Low Cost Nano System
5V	V _{cc}
GND	GND
SIG	P3.2

Tabel 2

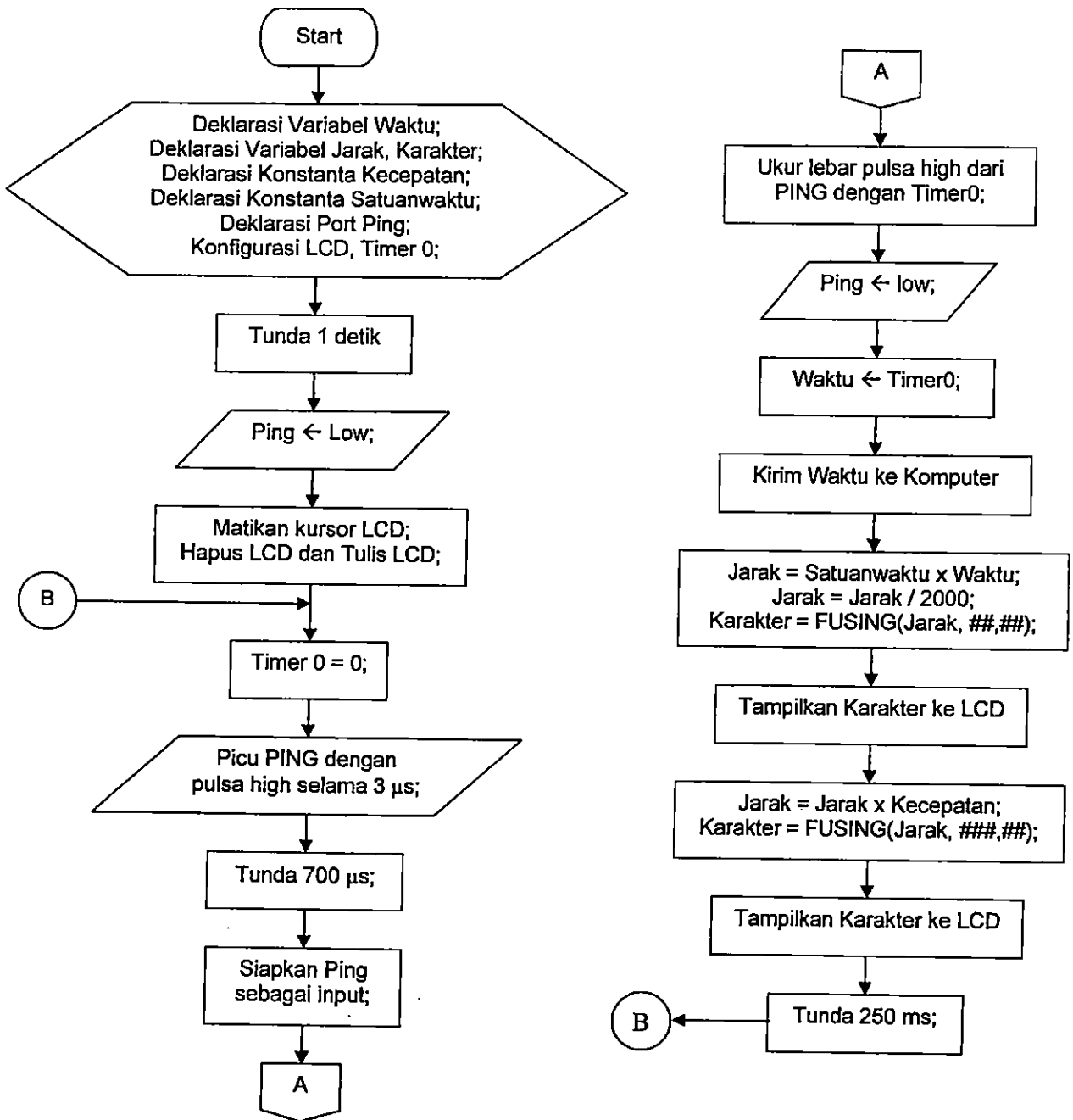
Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan Sensor PING)))™

Pin LCD yang tidak tertulis pada Tabel 1 tidak digunakan, karena menggunakan komunikasi data 4-bit. Sedangkan pin LCD yang diberi tanda * (V_o) dihubungkan ke VR 5k ohm sesuai Gambar 3 untuk pengaturan kontras.

Hubungkan modul-modul sesuai pada Tabel 1 dan 2, dan juga hubungkan port serial UART DT-51™ Low Cost Micro System / Low Cost Nano System ke port serial UART COM1 komputer.

Jika menggunakan DT-51 Low Cost Nano System, programlah terlebih dulu Rng_Find_v3.hex ke chip AT89C2051 menggunakan DT-HiQ Programmer atau DT-51 MinSys ver 3.0 + DT-51 ProgPAL atau parallel programmer lain. Untuk DT-51 Low Cost Micro System, pemrograman ke chip AT89S51 dapat dilakukan secara on-system menggunakan AT89S ISP Programmer Cable.

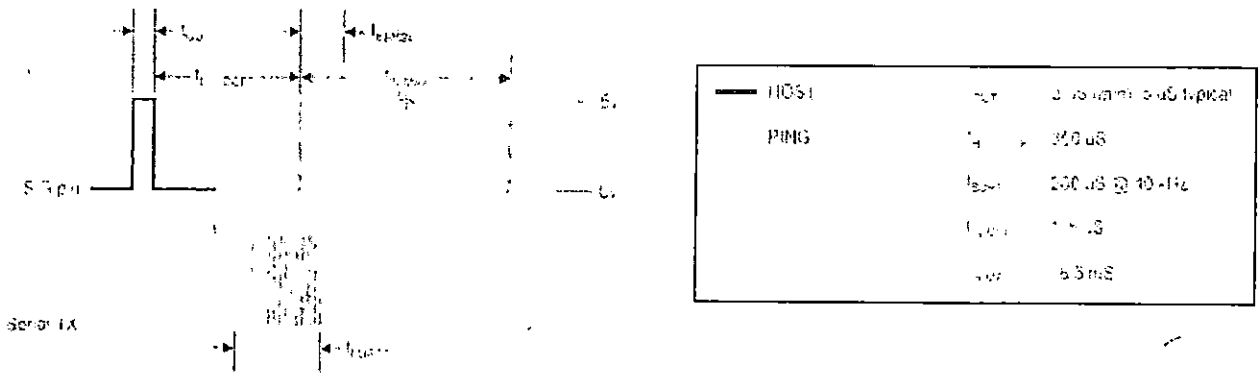
Flowchart dari sistem ini adalah sebagai berikut:



Gambar 6
Flowchart Program untuk DT-51™ Low Cost Micro System / Low Cost Nano System

Program mikrokontroler akan diproses sebagai berikut:

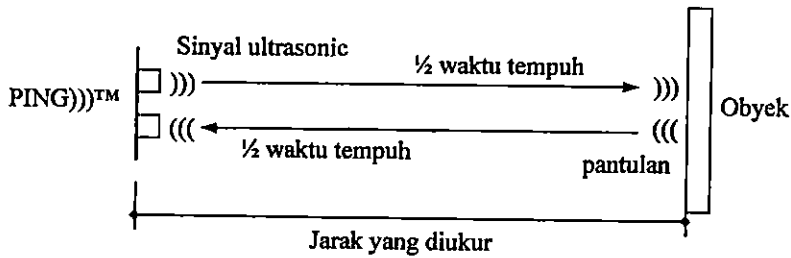
1. Proses yang pertama dilakukan adalah: deklarasi variabel Waktu, Jarak, Karakter; deklarasi konstanta Kecepatan, Satuanwaktu; deklarasi port Ping; konfigurasi terhadap Timer0 dan LCD.
 - Waktu adalah variabel bertipe word, berisi waktu tempuh untuk 2 kali jarak yang diukur (dalam satuan siklus mesin).
 - Jarak adalah variabel bertipe single, digunakan untuk data waktu pantul dalam ms dan data jarak yang diukur dalam satuan cm.



Gambar 4
Diagram waktu sensor PING)))™

Teori operasi PING)))™:

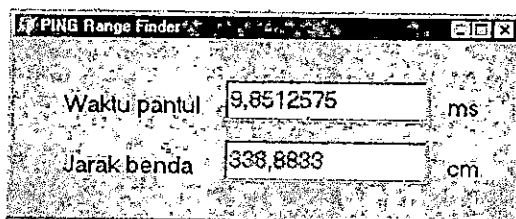
Sensor Ping mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama t_{BURST} (200 μ s) kemudian mendeteksi pantulannya. Sensor Ping memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan t_{OUT} min. 2 μ s). Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. Ping mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi Ping akan membuat *output low* pada pin SIG. Lebar pulsa High (t_{NH}) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur adalah $[(t_{NH} \text{ s} \times 344 \text{ m/s}) + 2]$ meter.



Gambar 5
Gambar Ilustrasi Cara Kerja PING)))™

- Karakter adalah variabel bertipe string dengan panjang 6 karakter, digunakan untuk data waktu tempuh dan jarak yang diukur dalam format string.
 - Kecepatan adalah konstanta, berisi kecepatan suara di udara yaitu 34.4 cm/ms.
 - Satuanwaktu adalah konstanta, berisi waktu satu siklus mesin yaitu 1.085 μ s untuk kristal 11.0592 MHz.
 - Ping adalah I/O port untuk sensor PING)))™, yaitu P3.2
 - Timer/Counter0 diatur pada fungsi timer, mode 1 (16-bit), *gate* internal.
 - LCD diatur untuk display 24x2 agar BASCOM-8051® dapat menggunakan LCD.
2. Setelah itu program menunda selama 1 detik. Dan kemudian memberi P3.2 logika *low*, mematikan kursor LCD, membersihkan layar LCD, serta menulis "Waktu Pantul=" pada baris 1 LCD dan "Jarak Benda =" pada baris 2 LCD.
 3. Program memberi nilai nol pada *register* Timer0 dan memicu Ping dengan mengirim pulsa *trigger* (pulsa *high* pada P3.2 selama 3 μ s) agar PING)))™ memulai pengukuran jarak.
 4. Program memberi P3.2 logika *low* lalu menunda selama 700 μ s. Setelah itu program menyiapkan P3.2 sebagai *input* (P3.2 diberi logika *high*) dan menunggu lagi selama 3 μ s. Sehingga dapat menerima pulsa *output* dari PING)))™.
 5. Setelah itu program mengukur lebar pulsa *high* pada *output* sensor PING)))™ menggunakan Timer0. Caranya: pada saat P3.2 = *high* Timer0 diaktifkan sampai pada saat P3.2 = *low* Timer0 dimatikan. Isi *register* Timer0 menyatakan lama waktu tempuh suara untuk 2x jarak ukur dalam satuan siklus mesin mikrokontroler.
 6. Setelah itu program akan me-*reset* P3.2 dan menyimpan data dalam *register* Timer 0 ke variabel Waktu.
 7. Dan program mengirimkan data Waktu tersebut ke Komputer melalui port serial UART 8-bit data, 1-bit stop, dan tanpa *parity* dengan baudrate 9600 bps.
 8. Melakukan perhitungan waktu pantul berdasarkan data dalam variabel Waktu tersebut sebagai berikut:
 Jarak = Satuanwaktu x Waktu → variabel Jarak akan berisi data 2x waktu tempuh dalam μ s
 Jarak = Jarak / 2000 → variabel Jarak akan berisi data waktu pantul dalam ms
 Karakter = FUSING(Jarak, ##.##) → variabel Karakter akan berisi data waktu pantul yang diukur dalam format string
 FUSING berfungsi untuk membuat agar Karakter memiliki format: dua angka di depan koma dan dua angka di belakang koma.
 9. Kemudian program menampilkan data waktu pantul tersebut ke LCD.
 10. Melakukan perhitungan jarak benda berdasarkan data waktu pantul tersebut sebagai berikut:
 Jarak = Jarak x Kecepatan → variabel Jarak akan berisi data jarak benda dalam cm
 Karakter = FUSING(Jarak, ##.##) → variabel Karakter akan berisi data jarak benda yang diukur dalam format string
 11. Kemudian program menampilkan data jarak benda tersebut ke LCD.
 12. Program menunda 250 ms lalu memulai pengukuran baru (kembali ke langkah 3).

pada saat dieksekusi, program Project2.EXE yang disertakan akan membuka COM1 dan menerima data sebanyak byte dari mikrokontroler.



Gambar 7
Tampilan Program Project2.EXE

arena data yang diterima merupakan data mentah (data Timer0), maka program ini juga akan melakukan perhitungan yang sama dengan perhitungan di mikrokontroler dengan persamaan:

$$\text{Waktu} = \text{Data} \times 1.085 / 2000 \rightarrow \text{waktu pantul}$$

$$\text{Jarak} = \text{Waktu} \times 34.4 \rightarrow \text{jarak benda}$$

Waktu akan ditampilkan dengan satuan milidetik, sedangkan Jarak akan ditampilkan dengan satuan sentimeter.

saat program ditutup, program akan memutuskan hubungan dengan COM1.

odul PING)))™ ini menggunakan satu pin sebagai *input* dan *output*. Oleh karena itu, hal yang harus diperhatikan dalam menggunakan modul ini adalah penentuan saat yang tepat untuk mengatur *port* mikrokontroler sebagai

Application Note AN73

input setelah mengeluarkan pulsa trigger ke PING)))™. Bila terlalu cepat atau terlalu lambat mengatur port 3.2 sebagai *input* (diberi logika *high*) akan menyebabkan kesalahan pengukuran jarak, hasil pengukuran menjadi lebih jauh atau lebih dekat dari jarak sesungguhnya.

Keuntungan menggunakan modul PING)))™ ini yaitu pada sisi *hardware*. Modul PING)))™ tidak membutuhkan komponen tambahan, dan memiliki *output* digital, serta hanya memerlukan satu pin I/O sehingga menghemat pin mikrokontroler.

Posisi bidang benda yang akan diukur jaraknya harus tegak lurus terhadap garis pandang sensor, jika tidak maka akan terjadi pemantulan gelombang ultrasonik yang tidak sempurna dan menyebabkan kesalahan pengukuran.

Listing program terdapat pada **AN73.ZIP**.

Selamat berinovasi!

ASCOM-8051 is copyright by MCS Electronics.

PING))) is a trademark of Parallax, Inc.

UT-51 is a trademark of Innovative Electronics.

LAMPIRAN 4

FOTO ALAT

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

