

**REMOTE TELEMETRY UNIT (RTU) PADA SISTEM PEMANTAU
CUACA MENGGUNAKAN DIAL UP HANDPHONE SEBAGAI
MEDIA TRANSFER DATA**

SKRIPSI

**Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik**



**Disusun oleh:
AGUNG SETIABUDI
NIM. 0510630008-63**

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG**

2010

**REMOTE TELEMETRY UNIT (RTU) PADA SISTEM PEMANTAU
CUACA MENGGUNAKAN DIAL UP HANDPHONE SEBAGAI
MEDIA TRANSFER DATA**

SKRIPSI

*Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik*



**Disusun oleh :
AGUNG SETIABUDI
NIM. 0510630008-63**

Telah diperiksa dan disetujui oleh
Dosen Pembimbing:

Pembimbing I

Pembimbing II

Ir. Nanang Sulistiyanto

NIP. 19700113 199403 1 002

Panca Mudjirahardjo, ST., MT.

NIP. 19700329 20012 1 001

LEMBAR PENGESAHAN

**REMOTE TELEMETRY UNIT (RTU) PADA SISTEM PEMANTAU
CUACA MENGGUNAKAN DIAL UP HANDPHONE SEBAGAI
MEDIA TRANSFER DATA**

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik

Disusun oleh:

**AGUNG SETIABUDI
NIM. 0510630008-63**

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 28 Januari 2010

DOSEN PENGUJI

Ir. Nurussa'adah, MT
NIP. 19680706 199203 2 001

Ir. M. Julius St., MS
NIP. 19540720 198203 1 002

M. Rif'an, ST., MT
NIP. 19710301 200012 1 001

Mengetahui
Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., M.Sc
NIP. 19710615 199802 1 003

Pengantar

Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, taufik dan hidayah-Nya lah skripsi ini dapat diselesaikan.

Skripsi berjudul “*Remote Telemetry Unit (RTU) pada Sistem Pemantau Cuaca Menggunakan Dial Up Handphone Sebagai Media Transfer Data*” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Ibu dan Ayah atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini,
- Rudy Yuwono, ST., M.Sc selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Ir. M. Julius St, MS selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya,
- Ir. Nanang Sulistyanto dan Panca Mudjirahardjo ST., MT. selaku Dosen Pembimbing 1 dan Dosen Pembimbing 2 atas segala bimbingan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan,
- Staf rekording Jurusan Teknik Elektro,
- Teman-teman Elkamania, Streamline, teman-teman di kelembagaan, senior serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan namanya satu-persatu, terima kasih banyak atas bantuan dan dukungannya.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, Januari 2010

Penulis

Abstrak

Agung Setiabudi, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Januari 2010, *Remote Telemetry Unit (RTU) pada Sistem Pemantau Cuaca Menggunakan Dial Up Handphone Sebagai Media Transfer Data*, Dosen Pembimbing: Ir. Nanang Sulistyanto dan Panca Mudjirahardjo ST., MT.

Indonesia adalah negara beriklim tropis yang mempunyai dua musim yaitu musim hujan dan musim kemarau. Pada beberapa tahun terakhir ini kondisi iklim dan cuaca di Indonesia menjadi tidak menentu dan perubahannya sulit untuk diprediksi atau diramalkan. Kondisi iklim dan cuaca yang tidak menentu sangat membingungkan masyarakat awam pada umumnya terutama masyarakat yang memiliki mata pencaharian yang erat kaitannya dengan iklim dan cuaca. Oleh karena itu perlu dibuat sebuah sistem pemantau cuaca yang dapat melakukan pemantauan dan pencatatan kondisi cuaca secara aktual serta dapat diakses dari jarak jauh.

Maka dalam skripsi ini akan dirancang Sistem Pemantau Cuaca Menggunakan *Dial up Handphone* Sebagai Media Transfer Data. Sistem ini terdiri atas Remote Telemetry Unit (RTU) dan Base Station. RTU merupakan suatu peralatan yang berada di tempat yang sama dengan besaran yang diamati, dan bertugas mengirimkan data hasil pengamatan ke *Base Station*. Sedangkan *Base Station* adalah unit yang berada di tempat yang terpisah dengan besaran yang diamati, dan bertugas menerima dan mengolah data yang diterima dari RTU.

RTU dapat megakuisisi data-data parameter cuaca yang meliputi suhu udara dengan resolusi 1 °C, kelembaban dengan resolusi 1 %RH, kecepatan angin dengan resolusi 1 km/jam, arah angin dengan resolusi 45° dan curah hujan dengan resolusi 1 mm. Data-data yang diakuisisi RTU ini kemudian ditransmisikan ke *Base Station* melalui kanal suara handphone yang didahului proses dial up dengan kecepatan maksimum 14 detik dalam satu kali transfer data. Data-data yang diterima *Base Station* kemudian diolah dan disimpan ke dalam *database* serta ditampilkan dalam bentuk grafik untuk mempermudah antarmuka dengan pengguna data.

Kata Kunci: RTU, pemantau cuaca, *dial up*

Daftar Isi

Pengantar.....	i
Abstrak	ii
Daftar Isi.....	iii
Daftar Gambar	v
Daftar Tabel	vii
BAB I PENDAHULUAN	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Cuaca	5
2.2 Iklim.....	5
2.3 Suhu Udara.....	6
2.4 Kelembaban.....	6
2.5 Sensor Suhu dan Kelembaban SHT11.....	7
2.6 Kecepatan dan Arah Angin.....	8
2.6.1 Sensor Kecepatan Angin (<i>Anemometer Cup</i>)	8
2.6.2 Sensor Arah Angin	9
2.7 Curah Hujan.....	9
2.8 Mikrokontroler ATmega162	11
2.9 Pin Out Handphone (Siemens C45).....	15
2.10 AT Command	16
2.11 Dual-Tone Multi-Frequency (DTMF).....	17
2.12 DTMF Decoder IC MT8870	18
2.13 Metode Deteksi Kesalahan <i>Checksum</i>	24
BAB III METODOLOGI PENELITIAN	25
3.1 Perancangan Alat	25
3.2 Pembuatan Alat.....	25
3.3 Pengujian Alat.....	25
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	28
4.1 Penentuan Spesifikasi Alat.....	28
4.2 Diagram Blok Sistem.....	28
4.3 Perancangan Mekanik	30
4.4 Perancangan Perangkat Keras (<i>Hardware</i>).....	33

4.4.1 Perancangan Rangkaian Opto Untuk Pemodelan Sensor Kecepatan dan Arah Angin	33
4.4.2 Perancangan Rangkaian Antarmuka SHT11 Dengan Mikrokontroler	35
4.4.3 Perancangan rangkaian antarmuka LCD LMB162ADC	36
4.4.4 Perancangan Rangkaian Antarmuka <i>Handphone</i> Dengan Mikrokontroler ...	37
4.4.5 Perancangan Rangkaian DTMF <i>Decoder</i>	38
4.5 Perancangan Format Data	40
4.6 Perancangan Protokol Komunikasi	41
4.7 Perancangan Perangkat Lunak Mikrokontroler	42
4.7.1 Perancangan Sub Rutin Mengubah Nilai Integer ke ASCII	42
4.7.2 Perancangan Sub Rutin Mengirim Data	43
4.7.3 Perancangan Sub Rutin Menerima Panggilan	44
4.7.4 Perancangan Sub Rutin Mengambil Data SHT11	44
4.7.5 Perancangan Sub Rutin <i>Interrupt</i> RX	45
4.7.6 Perancangan Sub Rutin Menampilkan Data pada LCD	46
4.7.7 Perancangan Sub Rutin Pengecekan Koneksi <i>Handphone</i>	46
4.7.8 Perancangan Sub Rutin Mengambil Data Arah Angin	46
4.7.9 Perancangan Sub Rutin Pengecekan <i>Limit Switch</i>	47
4.7.10 Perancangan Sub Rutin <i>Interrupt</i> Timer 2	49
4.7.11 Perancangan Rutin Utama	49
BAB V PENGUJIAN DAN ANALISIS	52
5.1 Pengujian Modul LCD 16x2	52
5.2 Pengujian Sensor	53
5.2.1 Pengujian Penghitungan Penekanan <i>limit switch</i>	53
5.2.2 Pengujian Rangkaian <i>Opto</i>	54
5.2.3 Pengujian modul DT-Sense SHT11	58
5.2.4 Pengujian Pemodelan Sensor Arah angin	60
5.3 Pengujian Pengiriman Data Secara Serial	61
5.4 Pengujian Antarmuka <i>Handphone</i> Dengan Mikrokontroler	62
5.5 Pengujian Penerjemahan Sinyal DTMF yang Dibangkitkan dan Ditransmisikan Melalui Kanal Suara <i>Handphone</i>	63
5.6 Pengujian Sistem Secara Keseluruhan	65
BAB VI KESIMPULAN DAN SARAN	69
6.1 Kesimpulan	69
6.2 Saran	70
Daftar Pustaka	71

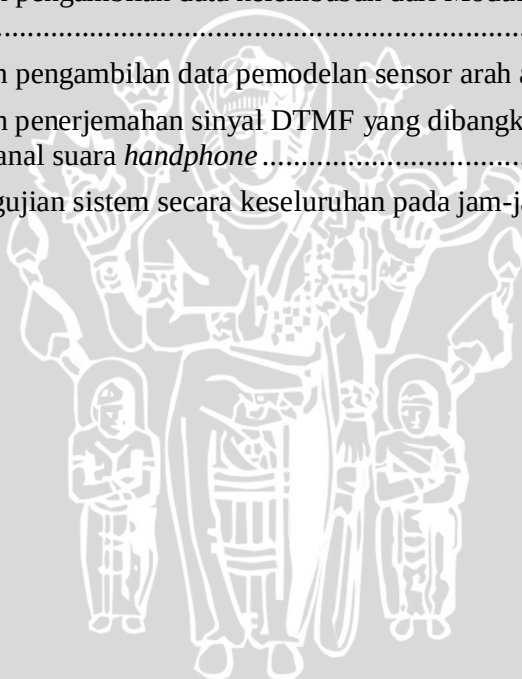
Daftar Gambar

Gambar 2.1 Sensor Temperatur dan Kelembaban SHT11	7
Gambar 2.2 Pengantarmukaan SHT11 dengan mikrokontroler.....	7
Gambar 2.3 Anemometer cup	8
Gambar 2.4 Sensor arah angin	9
Gambar 2.5 Tipping bucket	10
Gambar 2.6 Konfigurasi Pin ATmega162.....	11
Gambar 2.7 Format Pengiriman Data Serial USART ATmega162	14
Gambar 2.8 Pin <i>out</i> Siemens C45	15
Gambar 2.9 Frekuensi <i>Keypad DTMF</i>	18
Gambar 2.10 Konfigurasi Pin <i>Decoder DTMF</i> IC MT8870	19
Gambar 2.11 Diagram Blok Fungsi <i>Decoder DTMF</i> IC MT8870.....	20
Gambar 2.12 Diagram Waktu Penerima <i>DTMF</i> IC MT8870.....	22
Gambar 4.1 Diagram blok sistem secara keseluruhan.	29
Gambar 4.2 Rancangan mekanik pemodelan sensor kecepatan dan arah angin.....	30
Gambar 4.3 Piringan yang digunakan untuk mendeteksi arah angin.....	31
Gambar 4.4 Hasil pembuatan mekanik pemodelan sensor kecepatan angin, arah angin dan curah hujan	33
Gambar 4.5 Rangkaian opto yang terdiri atas LED dan <i>photodiode</i>	34
Gambar 4.6 Modul DT-Sense SHT11	35
Gambar 4.7 Rangkaian antarmuka DT-Sense SHT11 dengan mikrokontroler ATmega162	36
Gambar 4.8 Rangkaian LCD LMB162ADC	36
Gambar 4.9 <i>Interface Handphone</i> Siemens C45 ke Mikrokontroler ATmega162.....	37
Gambar 4.10 Hasil pembuatan minimum sistem ATmega162	38
Gambar 4.11 Konfigurasi rangkaian <i>DTMF decoder</i>	40
Gambar 4.12 Hasil pembuatan rangkaian <i>DTMF decoder</i>	40
Gambar 4.13 Format data komunikasi antara mikrokontroler ATmega162 dengan komputer <i>Base Station</i>	40
Gambar 4.14 Diagram alir sub rutin mengubah nilai integer ke ASCII.....	43
Gambar 4.15 Diagram alir sub rutin mengirim data ini ditunjukkan dalam.....	43
Gambar 4.16 Diagram alir sub rutin menerima panggilan	44
Gambar 4.17 Diagram alir sub rutin mengambil data SHT11	44
Gambar 4.18 Diagram alir sub rutin <i>interrupt RX</i>	45
Gambar 4.19 Diagram alir sub rutin menampilkan data pada LCD	46
Gambar 4.20 Diagram alir subrutin pengecekan koneksi <i>handphone</i>	47
Gambar 4.21 Diagram alir subrutin mengambil data arah	48

Gambar 4.22 Diagram alir sub rutin pengecekan <i>limit switch</i>	49
Gambar 4.23 Diagram alir sub rutin <i>interrupt</i> Timer 2.....	50
Gambar 4.24 Diagram alir rutin utama.....	51
Gambar 5.1 Diagram blok pengujian modul LCD.....	52
Gambar 5.2 Tampilan hasil pengujian modul LCD.....	53
Gambar 5.3 Diagram blok pengujian penghitungan penekanan <i>limit switch</i>	53
Gambar 5.4 Hasil pengujian penghitungan penekanan <i>limit switch</i> sebanyak 20 kali ...	54
Gambar 5.5 Hasil pengujian penghitungan penekanan <i>limit switch</i> sebanyak 300 kali..	54
Gambar 5.6 Diagram blok pengujian kecepatan respons sensor opto dalam	55
Gambar 5.7 Respons keluaran rangkaian opto terhadap masukan pada tepi naik.....	55
Gambar 5.8 Respons keluaran rangkaian opto terhadap masukan pada tepi turun.....	55
Gambar 5.9 Respons keluaran rangkaian opto terhadap masukan pada frekuensi 100,4 Hz	56
Gambar 5.10 Diagram blok pengujian jumlah pulsa rangkaian <i>opto</i>	57
Gambar 5.11 Hasil pengujian pemutaran piringan sebanyak lima kali.....	57
Gambar 5.12 Diagram blok pengujian modul DT-Sense SHT11	58
Gambar 5.13 Hasil pengujian pengambilan data suhu dari Modul DT-Sense SHT11 ...	59
Gambar 5.14 Hasil pengujian pengambilan data kelembaban dari Modul DT-Sense SHT11.....	59
Gambar 5.15 Diagram blok pengujian modul DT-Sense SHT11	60
Gambar 5.16 Hasil pengujian pengambilan data pemodelan sensor arah angin.....	61
Gambar 5.17 Diagram blok pengujian modul LCD.....	61
Gambar 5.18 Tampilan hasil pengujian pengiriman data secara serial dalam terminal komputer	62
Gambar 5.19 Diagram blok pengujian antarmuka <i>handphone</i> dengan mikrokontroler..	62
Gambar 5.20 Hasil pengujian antarmuka <i>handphone</i> dengan mikrokontroler.....	63
Gambar 5.21 Diagram blok pengujian penerjemahan sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara <i>handphone</i>	64
Gambar 5.22 Diagram blok pengujian sistem secara keseluruhan	65
Gambar 5.23 Tampilan nilai data yang dikirim dari RTU ke <i>Base Station</i>	66
Gambar 5.24 Tampilan data yang diterima oleh <i>Base Station</i>	66
Gambar 5.25 Data dari RTU yang berhasil disimpan dalam basis data.....	67

Daftar Tabel

Tabel 2.1 Alamat <i>Vector Interrupt</i> dari ATMEGA 162.....	13
Tabel 2.2 Rumus Menghitung <i>Boudrate</i>	14
Tabel 2.3 Fungsi masing-masing pin pada Siemens C45.....	15
Tabel 2.4 <i>AT Command</i> yang berhubungan dengan <i>dial up</i>	17
Tabel 2.5 Respons komunikasi data.....	17
Tabel 2.6 Tabel Fungsi <i>Decoder DTMF</i> IC MT8870	21
Tabel 4.1 Kombinasi logika untuk masing-masing arah angin	32
Tabel 5.1 Hasil pengujian penekanan limit switch	54
Tabel 5.2 Hasil pengujian penghitungan pulsa pada pemodelan sensor kecepatan angin.....	58
Tabel 5.3 Hasil pengujian pengambilan data suhu dari Modul DT-Sense SHT11	59
Tabel 5.4 Hasil pengujian pengambilan data kelembaban dari Modul DT-Sense SHT11	60
Tabel 5.4 Hasil pengujian pengambilan data pemodelan sensor arah angin	61
Tabel 5.5 Hasil pengujian penerjemahan sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara <i>handphone</i>	64
Tabel 5.6 Data hasil pengujian sistem secara keseluruhan pada jam-jam tertentu	67



BAB I PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah negara beriklim tropis yang mempunyai dua musim yaitu musim hujan dan musim kemarau. Pada beberapa tahun terakhir ini kondisi iklim dan cuaca di Indonesia menjadi tidak menentu dan perubahannya sulit untuk diprediksi atau diramalkan. Kondisi iklim dan cuaca yang tidak menentu sangat membingungkan masyarakat awam pada umumnya terutama masyarakat yang memiliki mata pencaharian yang erat kaitannya dengan iklim dan cuaca.

Selama ini pemantauan terhadap cuaca di Indonesia kurang optimal karena terhambat oleh beberapa kendala yaitu tidak semua wilayah Indonesia berada dalam jangkauan stasiun cuaca akibat keterbatasan jumlah peralatan karena harganya yang mahal. Selain itu tidak semua stasiun pemantau cuaca beroperasi secara otomatis sehingga proses transfer data ke stasiun pusat masih berjalan lambat.

Proses transfer data parameter cuaca yang terpantau pada stasiun cuaca yang beroperasi secara manual dilakukan dengan cara mencatat setiap parameter cuaca yang terekam oleh petugas pencatat. Data tersebut kemudian dikirimkan ke stasiun pusat pemantau cuaca. Sistem konvensional ini memiliki beberapa kekurangan yaitu dimungkinkan terjadinya kesalahan petugas (*human error*) serta data tidak dapat dikirimkan dalam jangka waktu yang singkat.

Di sisi lain ketersediaan informasi tentang kondisi cuaca dengan cepat dan akurat menjadi suatu hal yang penting bagi masyarakat yang memiliki tingkat ketergantungan yang tinggi pada kondisi iklim dan cuaca terutama masyarakat yang memiliki mata pencaharian sebagai petani maupun nelayan. Selain itu informasi cuaca juga diperlukan oleh bidang usaha yang berhubungan dengan sektor pertanian dan perikanan.

Oleh karena itu perlu dibuat sebuah sistem pemantau cuaca yang dapat melakukan pemantauan dan pencatatan kondisi cuaca secara aktual serta dapat diakses dari jarak jauh. Maka dalam skripsi ini akan dirancang Sistem Pemantau Cuaca Menggunakan *Dial up Handphone* Sebagai Media Transfer Data. Sistem ini terdiri atas Remote Telemetry Unit (RTU) dan Base Station. RTU merupakan suatu peralatan yang berada di tempat yang sama dengan besaran yang diamati, dan bertugas mengirimkan data hasil pengamatan ke *Base Station*. Sedangkan *Base Station* adalah unit yang berada

di tempat yang terpisah dengan besaran yang diamati, dan bertugas menerima dan mengolah data yang diterima dari RTU. Umumnya, sistem telemetri menggunakan beberapa RTU dan sebuah *Base Station*.

Transmisi data dari RTU ke *base Station* dapat menggunakan berbagai media, diantaranya kabel, *Short Message Service* (SMS), *dial up* dan WIFI. Dalam skripsi ini media yang digunakan adalah *dial up* pada *handphone*. Keuntungan menggunakan *dial up* pada *handphone* sebagai media komunikasi data adalah daerah jangkauannya luas, sehingga antara RTU dan *Base Station* bisa terpisah dengan jarak yang jauh. Selain itu dengan menggunakan media ini biaya yang digunakan relatif murah karena hampir semua operator GSM di Indonesia memberlakukan tarif telepon hanya untuk satu menit pertama dan 60 menit berikutnya hampir tanpa biaya.

Sebagai antarmuka dengan pengguna, Sistem Pemantau Cuaca Terpusat Berbasis Telemetri ini dilengkapi dengan fasilitas SMS. Pengguna sistem ini dapat dengan mudah mengetahui kondisi cuaca di suatu wilayah yang telah dipasang RTU hanya dengan mengirim SMS dengan format dan nomor yang telah ditentukan sebelumnya.

Skripsi ini membahas tentang perancangan dan pembuatan RTU pada sistem pemantau cuaca menggunakan *dial up handphone* sebagai media transfer data. RTU ini terdiri dari sensor suhu dan kelembaban, sensor kecepatan angin, sensor arah angin, sensor curah hujan dan detektor hujan. Selain itu RTU ini juga terdiri dari beberapa rangkaian elektronika untuk mengkondisikan sinyal, mikrokontroler untuk memproses data dan modul GSM (*handphone*) untuk mentransmisikan data.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat disusun rumusan masalah sebagai berikut:

- 1) Bagaimana merancang dan membuat sistem elektronika untuk proses akuisisi data dari sensor suhu dan kelembaban, sensor arah angin, sensor kecepatan angin dan sensor curah hujan.
- 2) Bagaimana merancang dan membuat rangkaian antarmuka mikrokontroler dengan *handphone* dan sensor-sensor yang digunakan.
- 3) Bagaimana merancang dan membuat perangkat lunak sistem mikrokontroler sebagai pengolah data dan media kendali utama sistem transmisi data menggunakan *dial up* pada *handphone*.

1.3 Tujuan

Tujuan penyusunan skripsi ini adalah merealisasikan sebuah *Remote Telemetry Unit* (RTU) yang dapat mengamati besaran-besaran cuaca antara lain suhu, kelembaban, arah angin, kecepatan angin dan curah hujan dan mengirimkan data besaran-besaran tersebut ke Base Station. Dengan demikian dapat mempermudah masyarakat untuk mengetahui kondisi cuaca di daerah tertentu.

1.4 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan alat akan diberi batasan sebagai berikut:

- 1) Parameter cuaca yang diukur adalah suhu, kelembaban, kecepatan angin, arah angin dan curah hujan.
- 2) Perancangan sensor arah angin, sensor kecepatan angin dan sensor curah hujan hanya dimodelkan sesuai dengan prinsip kerja masing-masing sensor.
- 3) Mikrokontroler yang digunakan adalah ATmega162.
- 4) Sistem menggunakan catu daya DC 5-12 volt.

1.5 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Metodologi

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian, dan analisis data.

BAB IV Perancangan

Perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan realisasi alat.

BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar

terhadap hasil-hasil pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian dimasa yang akan datang.

UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

Dalam merencanakan dan merealisasikan sebuah *Remote Telemetry Unit* (RTU), maka dibutuhkan pemahaman tentang berbagai hal yang mendukung sistem ini. Pemahaman ini akan bermanfaat untuk merancang perangkat keras dan perangkat lunak sistem. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi iklim, cuaca dan parameter-parameternya, sensor suhu dan kelembaban SHT11, sensor kecepatan angin, sensor arah angin, sensor curah hujan, mikrokontroler ATmega162, komunikasi serial, DTMF dan AT Command.

2.1 Cuaca

Cuaca adalah keadaan atmosfer pada tempat tertentu pada waktu tertentu atau dalam periode tertentu, dengan perhatian khusus pada perubahan dalam jangka waktu yang singkat (Allaby, 2007: 528).

Dengan kata lain cuaca adalah keadaan atmosfer di suatu tempat yang selalu berubah tiap waktu. Keadaan ini meliputi suhu udara, kelembaban udara, curah hujan, angin dan unsur cuaca lainnya. Untuk mengetahui nilai masing-masing parameter tersebut biasanya dilakukan pengukuran atau pemantauan disuatu tempat. Pada tempat-tempat pengukuran tersebut biasanya didirikan sebuah stasiun cuaca. Stasiun cuaca adalah sebuah tempat yang dilengkapi dengan instrumen-instrumen yang dibutuhkan untuk melakukan standarisasi pengukuran dan observasi kondisi cuaca dan fasilitas komunikasi untuk mengirimkan laporan cuaca kepada stasiun pusat (Allaby, 2007: 542).

Di Indonesia sendiri untuk saat ini telah memiliki 173 stasiun cuaca yang tersebar di berbagai daerah di Indonesia. Jumlah ini sangat kurang mengingat wilayah indonesia yang sangat luas dan terdiri dari banyak pulau. Untuk itu perlu didirikan stasiun cuaca yang lebih banyak lagi.

2.2 Iklim

Iklim adalah rata-rata kondisi cuaca dalam waktu yang panjang, termasuk musim dan perubahannya, untuk daerah dan selang waktu tertentu. Iklim sering direpresentasikan sebagai sebuah kumpulan statistik dari kondisi cuaca untuk daerah dengan gambaran kondisi yang terbatas (waktu, musim, daerah) (Smith, 2006: 43).

Ada banyak iklim di dunia ini, tergantung kondisi dan letaknya secara geografis.

Secara umum iklim dibagi menjadi enam:

- 1) Iklim hujan tropik: suhu terdingin $> 18^{\circ}\text{C}$.
- 2) Iklim kering
- 3) Iklim hangat: suhu terdingin antara -3° dan 18°C , suhu terhangat $> 18^{\circ}\text{C}$.
- 4) Iklim dingin: suhu terdingin $< -3^{\circ}$, suhu terhangat $> 10^{\circ}\text{C}$
- 5) Iklim tundra: Suhu terhangat 0 sampai 10°C
- 6) Iklim beku: suhu terhangat $< 0^{\circ}\text{C}$.

Lima dari enam tipe iklim utama didasarkan pada nilai temperatur bulanan (Barry dan Chorley, 2003: 391).

2.3 Suhu Udara

Suhu adalah derajat panas sebuah bahan (padat, cair atau gas), yang berhubungan dengan energi kinetik rata-rata atom-atom atau molekul bahan. Suhu merupakan elemen yang sangat penting dalam klimatologi dan meteorologi, suhu dicatat pada stasiun cuaca melalui termometer yang diletakkan dalam sebuah tempat instrumen yang terlindungi; Thermistor digunakan dalam stasiun cuaca otomatis dan pada level yang lebih tinggi dalam pengukuran *radiosonde*. Untuk menyatakan skala atau nilai suhu secara umum dalam meteorologi digunakan skala *celsius*, meskipun skala fahrenheit masih sering digunakan di Amerika (Smith, 2006: 214).

Suhu udara yang terukur dalam alat ukur adalah suhu udara pada saat itu saja, sehingga untuk mengetahui suhu udara harian, nilai-nilai suhu udara hasil pengamatan yang dilakukan selama satu hari harus dirata-rata.

2.4 Kelembaban

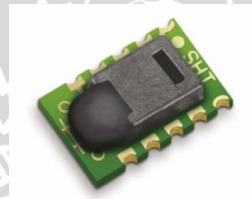
Kelembaban adalah banyak sedikitnya uap air di udara. Kelembaban dapat dinyatakan dalam beberapa cara yaitu:

- 1) Kelembaban absolut (*absolute humidity*), yaitu bilangan yang menunjukkan berat uap air tiap kesatuan volume udara yang dinyatakan dalam gram uap air per m^3 Udara.
- 2) Kelembaban spesifik (*specific humidity*), berat uap air tiap kesatuan berat udara biasanya dinyatakan dalam gram uap air per kg udara.
- 3) Tekanan uap (*vapor pressure*), yaitu menyatakan besarnya tekanan yang diberikan oleh uap air sebagai bagian dari udara.

- 4) Kelembaban relatif (*relative humidity*), perbandingan antara banyaknya air yang terdapat di udara dengan banyaknya uap air maksimum yang dapat dikandung oleh udara pada suhu dan tekanan yang sama. Kelembaban relatif dinyatakan dalam % (Waryono, 1987: 58).

2.5 Sensor Suhu dan Kelembaban SHT11

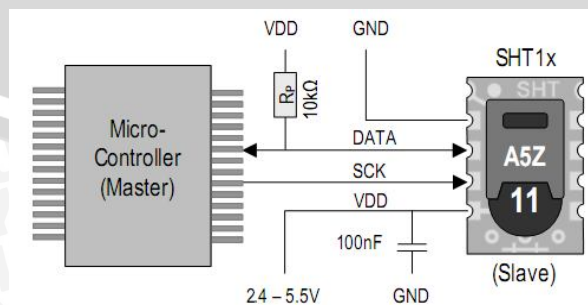
SHT11 ialah modul sensor digital untuk temperatur sekaligus kelembaban pertama di dunia yang diklaim oleh pabrik pembuatnya, Sensirion Corp. Sensor ini telah dikalibrasikan secara standar oleh pabrik pembuatnya dan telah mempunyai keluaran berupa data digital. Modul sensor ini tidak memerlukan lagi perangkat tambahan luar untuk pengantarmukaannya sehingga mempermudah dalam pengaplikasiannya terutama ketika dihubungkan secara langsung dengan mikrokomputer, mikroprosesor dan sejenisnya. Gambar 2.1 menunjukkan modul sensor SHT11.



Gambar 2.1 Sensor Temperatur dan Kelembaban SHT11

Sumber: Sensirion, 2008: 1

Sensor ini mempunyai kisaran pengukuran dari 0-100% RH, dan akurasi RH relatif $\pm 3\%$ RH. Sedangkan akurasi pengukuran temperatur $\pm 0.4^\circ\text{C}$. Konsumsi daya dari sensor ini sangatlah kecil. Sensor ini bekerja dengan *interface 2-wire*, maksudnya adalah hanya membutuhkan 2 buah jalur sebagai komunikasi datanya dengan mikrokontroler, seperti ditunjukkan dalam Gambar 2.2. Aplikasi sensor ini antara lain pada stasiun cuaca, data *logging*, pemancar, otomotif dan berbagai perangkat instrumentasi lain.



Gambar 2.2 Pengantarmukaan SHT11 dengan mikrokontroler

Sumber: Sensirion, 2008: 5

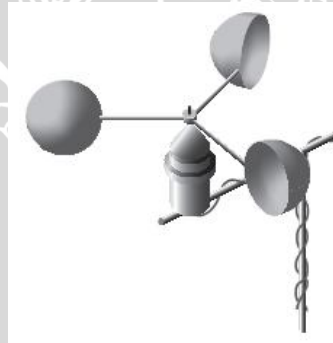
2.6 Kecepatan dan Arah Angin

Angin adalah perpindahan udara sebagai akibat dari ketidakrataan distribusi tekanan udara. Udara cenderung bergerak dari daerah bertekanan udara tinggi ke daerah bertekanan udara rendah dengan kecepatan yang sebanding dengan gradien tekanan (Allaby, 2007: 549).

Angin adalah besaran vektor, jadi dinyatakan dalam arah dan kecepatan. Secara klimatologi arah angin pada umumnya cukup diamati delapan arah saja, yaitu Utara (U), Timur Laut (TL), Timur (T), Tenggara (Tg), Selatan (S), Barat Daya (BD), Barat (B) dan Barat Laut (BL). Kecepatan angin dinyatakan dalam satu satuan berikut, yaitu m/det, km/jam dan knot, 1 knot $\approx$ 10,5 m/det.

2.6.1 Sensor Kecepatan Angin (*Anemometer Cup*)

Sebagai salah satu instrumen pada stasiun pengamat cuaca, anemometer adalah alat yang digunakan untuk mengukur kecepatan angin. Untuk mengukur kecepatan angin secara langsung sangat sulit, sehingga untuk mengukurnya digunakan volum udara dalam skala kecil dan melihat pergerakannya dalam jarak yang terukur. Bentuk anemometer ditunjukkan dalam Gambar 2.3.



Gambar 2.3 Anemometer cup

Sumber: Allaby, 2007: 27

Sebuah anemometer mencatat kecepatan angin pada waktu tertentu, tetapi alat ini juga bisa menentukan kecepatan angin rata-rata selama selang waktu tertentu. Jumlah putaran anemometer yang terjadi dalam selang waktu tertentu dapat dikonversi ke dalam jarak (D) yang dilalui sebuah *cup* dengan menggunakan persamaan:

$$D = \pi d R \quad (2-1)$$

Dimana d adalah diameter lingkaran yang dilalui *cup* dan R adalah jumlah putaran yang terjadi. Besaran-besaran ini dapat dikonversi menjadi kecepatan angin (S) dengan menggunakan rumus:

$$S = D/T$$

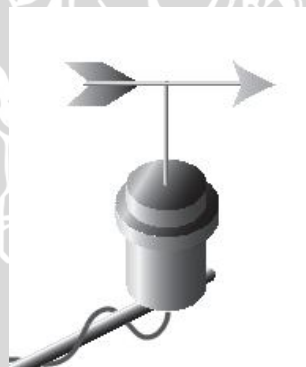
(2-2)

Dimana T adalah waktu yang dibutuhkan untuk melakukan R putaran (Allaby, 2007: 27-28).

Angin yang mengenai *Anemometer cup* akan menggerakkan *cup* tersebut sehingga poros anemometer berputar, hal ini akan menyebabkan piringan berputar. Bila gerakan angin semakin cepat, maka putaran anemometer akan semakin cepat pula. Jumlah putaran dapat dihitung dengan alat penghitung yang dipasang pada anemometer berupa *opto switch*. Sehingga alat ini dapat dimodelkan dengan sebuah *opto switch* dan sebuah piringan yang berputar.

2.6.2 Sensor Arah Angin

Untuk menentukan arah angin, digunakan penangkap angin yang dapat menunjukkan dari mana arah datangnya angin. Hal ini dapat diketahui dengan mengamati langsung alat tersebut. Untuk pemodelan sensor dapat digunakan beberapa LED, *photodiode* dan piringan yang telah diberi lubang dengan pola tertentu sehingga dengan mengetahui *photodiode* yang aktif dapat diketahui arah angin. Bentuk sensor arah angin ditunjukkan dalam Gambar 2.4.



Gambar 2.4 Sensor arah angin

Sumber: Allaby, 2007: 27

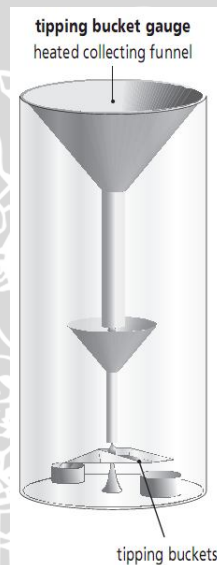
2.7 Curah Hujan

Hujan adalah *precipitation* dalam bentuk cair yang jatuh dengan ukuran diameter antara 0.02 inch sampai 0.2 inch (0.5–5.0 mm) (Allaby, 2007: 382). Sedangkan curah hujan adalah banyaknya hujan yang jatuh ke bumi per satuan luas. Banyaknya curah hujan yang per satuan waktu disebut dengan intensitas curah hujan (Smith, 2006: 185).

Untuk mengukur besarnya curah hujan digunakan sensor curah hujan. Sensor yang paling sering digunakan pada stasiun cuaca adalah sensor hujan dengan prinsip

tipping bucket. Sensor jenis ini menggunakan dua penampung kecil yang diletakkan di atas satu tumpuan. Kedua penampung dapat bergerak naik turun. Prinsip yang digunakan sensor jenis ini seperti permainan jungkat-jungkit. Apabila salah satu penampung penuh terisi air maka penampung tersebut akan turun dan penampung lainnya akan naik untuk diisi air.

Sensor curah hujan jenis tipping bucket digunakan bersamaan dengan corong pengumpul. Corong pengumpul diletakkan pada bagian atas dari sensor. Air yang terkumpul pada corong akan dialirkan ke penampung pada sensor. Apabila penampung penuh maka penampung akan turun sekaligus mengosongkan isi. Penampung lainnya akan naik dan siap untuk diisi. Pengambilan data dilakukan dengan cara menghitung berapa banyaknya kondisi naik-turun dalam satu periode tertentu. Gambar 2.5 menunjukkan bentuk fisik dari tipping bucket.



Gambar 2.5 Tipping bucket
Sumber: Allaby, 2007: 387

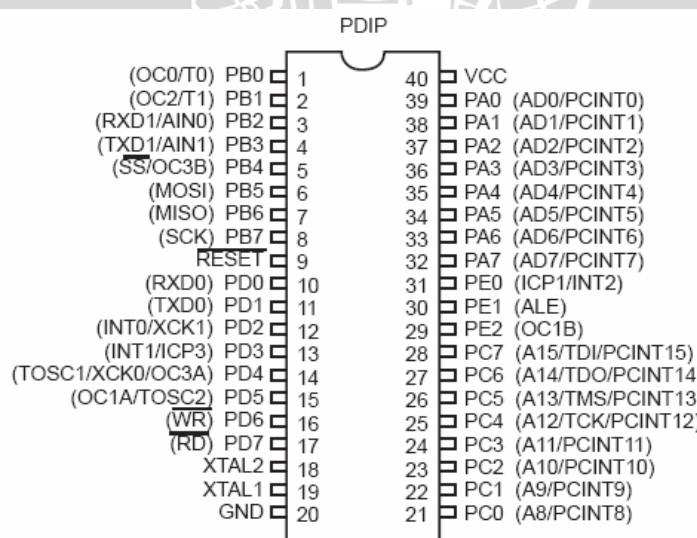
Banyaknya kondisi naik-turun dapat diukur secara elektronik dengan meletakkan sensor yang mampu mendeteksi pergerakan. Salah satu sensor yang biasa digunakan adalah *red switch*. *Red switch* mendeteksi pergerakan magnet yang terdapat disekitarnya. Dengan memasang magnet pada bagian yang bergerak akan dapat diukur banyaknya pergerakan pada tipping bucket ini. Sensor ini dapat dimodelkan dengan menggunakan sebuah *limit switch*.

2.8 Mikrokontroler ATmega162

Sebagai sentral/pusat dari pengontrolan sistem ini digunakan mikrokontroler ATmega162. Mikrokontroler ini yang bertugas membaca data dari sensor dan mengolah data tersebut menjadi format data yang siap dikirimkan serta mengirimkannya melalui *dial up* pada *handphone*. Mikrokontroler ATmega162 ini diproduksi oleh ATMEL Company Amerika Serikat dan merupakan salah satu anggota keluarga dari jenis AVR. Mikrokontroler ini memiliki fasilitas komunikasi serial untuk melakukan pengiriman/penerimaan data dengan *handphone* dan komputer. IC jenis ini berorientasi pada kontrol yang dapat diprogram ulang. Mikrokontroler ATmega162 mempunyai karakteristik utama sebagai berikut:

- CPU dengan lebar data 8 bit
- Empat buah I/O 8 bit dan sebuah I/O 3 bit.
- Memori Flash sebesar 16 kilo *byte*.
- SRAM sebesar 1 kilo *byte*.
- EEPROM sebesar 512 byte untuk memori program pada chip
- Dua buah timer/counter 8 bit dan 16 bit
- Enam buah saluran PWM
- Dua buah port USART untuk komunikasi serial
- Interupsi internal dan eksternal.
- Oscilator internal terdapat dalam chip

Konfigurasi pin mikrokontroler ATmega162 ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Konfigurasi Pin ATmega162

Sumber: Atmel, 2007: 2

Mikrokontroler ATmega162 memiliki 5 buah port. Empat buah port memiliki 8 buah jalur I/O dan sebuah port memiliki 3 buah jalur I/O yang bersifat bidirectional. Beberapa karakteristik port mikrokontroler ATmega162 dijelaskan secara singkat berikut ini:

- VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
- GND merupakan pin *ground*.
- Port A (PA0..PA7) merupakan pin I/O dua arah dan pin
- Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu SPI, Timer/Counter, komparator analog, dan USART1
- Port C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu JTAG, PCINT.
- Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, interupsi eksternal, dan USART0.
- Port E (PE0..PE7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, interupsi eksternal
- RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
- XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.

Mikrokontroler ATmega162 memiliki 28 alamat *vektor interrupt* dimana nomor urut dari *vector interrupt* tadi menyatakan prioritas dari *interrupt* tersebut. Alamat *vector interrupt* dari ATmega162 ditunjukkan dalam Tabel 2.1.

Mikrokontroler ATmega162 memiliki dua buah port USART untuk komunikasi serial, yaitu USART0 and USART1. USART0 dan USART1 digunakan untuk komunikasi antara ATmega162 dengan *handphone* dan transfer data ke *base station*. Fasilitas komunikasi serial USART mikrokontroler ini memiliki fitur sebagai berikut:

- Komunikasi full-duplex dengan register serial untuk penerima dan pengirim data.
- Dapat dioperasikan pada mode komunikasi sinkronus dan asinkronus.
- Pada operasi sinkronus clock berasal dari master atau slave.
- Mempunyai resolusi tinggi untuk generator boudrate.
- Layanan pengiriman data terdiri dari 5, 6, 7, 8 dan 9 bit dan 1 atau 2 bit stop.
- Paritas genap atau ganjil dan didukung dengan pengecekan paritas oleh hardware.
- Pendeteksi pengiriman kelebihan data.

- Pendeteksi kesalahan pada format data yang dikirim.
- Memiliki filter noise yang terdiri atas pendeteksi kesalahan bit start dan low pass filter.
- Memiliki 3 layanan *interrupt* yaitu TX complete, TX data empty, dan RX complete.
- Mode komunikasi multi processor.
- Mode komunikasi asinkron dengan dua kecepatan.

Tabel 2.1 Alamat Vector Interrupt dari ATMEGA 162

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	0x002	INT0	External Interrupt Request 0
3	0x004	INT1	External Interrupt Request 1
4	0x006	INT2	External Interrupt Request 2
5	0x008	PCINT0	Pin Change Interrupt Request 0
6	0x00A	PCINT1	Pin Change Interrupt Request 1
7	0x00C	TIMER3 CAPT	Timer/Counter3 Capture Event
8	0x00E	TIMER3 COMPA	Timer/Counter3 Compare Match A
9	0x010	TIMER3 COMPB	Timer/Counter3 Compare Match B
10	0x012	TIMER3 OVF	Timer/Counter3 Overflow
11	0x014	TIMER2 COMP	Timer/Counter2 Compare Match
12	0x016	TIMER2 OVF	Timer/Counter2 Overflow
13	0x018	TIMER1 CAPT	Timer/Counter1 Capture Event
14	0x01A	TIMER1 COMPA	Timer/Counter1 Compare Match A
15	0x01C	TIMER1 COMPB	Timer/Counter1 Compare Match B
16	0x01E	TIMER1 OVF	Timer/Counter1 Overflow
17	0x020	TIMER0 COMP	Timer/Counter0 Compare Match
18	0x022	TIMER0 OVF	Timer/Counter0 Overflow
19	0x024	SPI, STC	Serial Transfer Complete
20	0x026	USART0, RXC	USART0, Rx Complete
21	0x028	USART1, RXC	USART1, Rx Complete
22	0x02A	USART0, UDRE	USART0 Data Register Empty
23	0x02C	USART1, UDRE	USART1 Data Register Empty
24	0x02E	USART0, TXC	USART0, Tx Complete
25	0x030	USART1, TXC	USART1, Tx Complete
26	0x032	EE_RDY	EEPROM Ready
27	0x034	ANA_COMP	Analog Comparator
28	0x036	SPM_RDY	Store Program Memory Ready

Sumber: Atmel, 2007 : 58

Untuk menghitung *boudrate* dari komunikasi serial digunakan rumus seperti yang terlihat dalam Tabel 2.2.

Tabel 2.2 Rumus Menghitung Boudrate

Mode Operasi	Persamaan perhitungan Baud Rate	Persamaan perhitungan UBRR Rate
Asinkron :Normal mode	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asinkron : Double speed mode	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Sinkron : Master mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Sumber: Atmel, 2007: 171

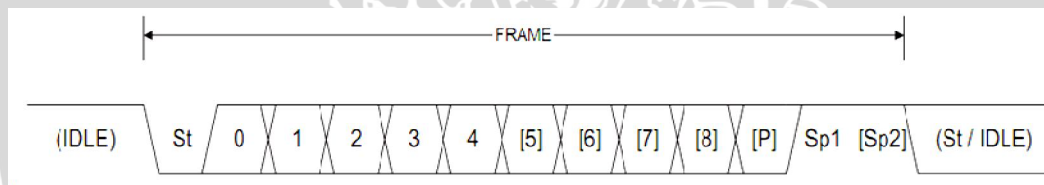
Keterangan:

f_{osc} = Frekuensi klok dari sistem osilator

UBRR = Register *baudrate* yang terdiri dari UBRRH dan UBRRH

BAUD = *Baudrate* dalam *bits per second* (bps)

Pada pengiriman data secara serial USART menggunakan ATmega162 memakai format seperti dalam Gambar 2.7.



Gambar 2.7 Format Pengiriman Data Serial USART ATmega162

Sumber: Atmel, 2007: 173

Dimana:

St = Bit start selalu berlogika rendah

P = Bit paritas (ganjil atau genap)

Sp = Bit stop selalu berlogika tinggi (bit stop bisa berjumlah 1 atau 2)

IDLE = Tidak ada data yang ditransfer pada RX dan TX, IDLE selalu berlogika teinggi

Perhitungan bit paritas dilakukan dengan cara menggunakan logika *exclusive-or* (XOR) terhadap semua bit-bit data yang ditransmisikan. Jika yang digunakan adalah paritas ganjil, maka hasil dari keluaran logika *exclusive-or* dinegasikan. Hubungan antara bit paritas dengan bit-bit data sesuai dengan persamaan berikut:

$$\begin{aligned}
 P_{even} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0 \\
 P_{odd} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1
 \end{aligned}
 \tag{2-3}$$

keterangan:

P_{even} : paritas yang digunakan adalah paritas genap.

P_{odd} : paritas yang digunakan adalah paritas ganjil.

d_n : bit data ke-n dari sebuah karakter.

2.9 Pin Out Handphone (Siemens C45)

Siemens C45 adalah salah satu model *handphone* yang dilengkapi dengan pin *out* yang berfungsi sebagai jalur berhubungan dengan perangkat luar seperti komputer atau mikrokontroler. Untuk dapat terhubung dengan mikrokontroler maka pin RX, TX dan GND pada *handphone* harus dihubungkan dengan pin TX, RX dan GND pada mikrokontroler.

Untuk mengirim dan menerima data dari perangkat luar siemens C45 dilengkapi dengan pin RX dan TX. Komunikasi yang terjadi adalah secara serial dengan baudrate 19200 bit/detik. Konfigurasi pin out dari siemens C45 ditunjukkan dalam Gambar 2.8.



Gambar 2.8 Pin out Siemens C45

Sumber: Pinouts, 2005: 1

Fungsi masing-masing pin pada siemens C45 secara lengkap ditunjukkan dalam Tabel 2.3.

Tabel 2.3 Fungsi masing-masing pin pada Siemens C45

Pin	Name	Dir	Description
1	GND	-	Ground
2	SELF-SERVICE	in/out	Pendeteksi/Pengontrol charger baterai
3	LOAD	in	Tegangan charging baterai
4	BATTERY	out	Battery
5	DATA OUT (TX)	out	Kirim data
6	DATA IN (RX)	in	Menerima data

7	Z_CLK	-	Jalur clock untuk <i>bus</i> tambahan. Digunakan sebagai DTR pada operasi data
8	Z_DATA	-	Jalur clock untuk <i>bus</i> tambahan. Digunakan sebagai CTS pada operasi data
9	MICG	-	Ground untuk microphone
10	MIC	in	Line masukan microphone (Vpp 1,5 volt)
11	AUD	out	Loudspeaker Headset (Vpp 1,5 volt)
12	AUDG	-	Ground untuk speaker tambahan

Sumber: Pinouts, 2005: 1

2.10 AT Command

Untuk memasukkan program ke perangkat *handphone*, tentunya perangkat tersebut harus mendukung untuk melakukan komunikasi dengan perangkat yang lain di luar sistem tersebut misalnya dengan PC maupun dengan *mikrokontroler*. Siemens C45 dilengkapi dengan kemampuan tersebut karena terdapatnya pin RX dan TX. Bahasa pemrograman yang digunakan untuk memprogram Siemens C45 atau yang lain dikenal dengan nama *AT-Command*.

AT-Command memiliki fungsi memberikan perintah kepada *handphone* untuk melakukan suatu tindakan berdasarkan pada perintah dari AT Command yang diberikan misalnya menghapus sms, menampilkan data sms, melakukan dial, dan lain-lain. Beberapa perintah *AT-Command* ditunjukkan dalam Tabel 2.4. Respons yang diberikan *handphone* ditunjukkan dalam Tabel 2.5.

Tabel 2.4 AT Command yang berhubungan dengan *dial up*

Command	Function
A/	Repeat last command
AT...	Prefix for all other commands
ATA	Accept call
ATD<str>;	Dial the dialing string <str> with the voice utility Valid dial modifiers: "T" (tone dialing), "P" (pulse dialing) is ignored. The character ";," is important, for this tells the phone that the call should be set up with the voice utility. Otherwise an attempt is made to set up a data call, which the phone immediately acknowledges with "ERROR". The dial command responds with OK to the user right after starting a voice call. Other behavior like *# sequences in the dial command and also data calls remain unchanged.
ATD><n>;	Dial the telephone number from the current telephone book location number <n> The telephone book is selected with the command at+cpbs (or at^spbs).
ATD><mem><n>;	Dial the telephone number from the telephone book <mem> location number <n>
ATDL	Dial last telephone number
ATE0	Deactivate command echo
ATE1	Activate command echo
ATH[0]	Separate connection
ATQ0	Display acknowledgments
ATQ1	Suppress acknowledgments
ATV0	Output acknowledgments as numbers
ATV1	Output acknowledgments as text
AT&F[0]	Reset to factory profile
ATZ	Set to default configuration
AT+GCAP	Output the capabilities list

Sumber: Siemens, 2000: 6

Tabel 2.5 Respons komunikasi data

Response	Numeric	Meaning
OK	0	Command executed, no errors
RING	2	Ring detected
NO CARRIER	3	Link not established or disconnected
ERROR	4	Invalid command or command line too long
NO DIALTONE	6	No dial tone, dialing impossible, wrong mode
BUSY	7	Remote station busy

Sumber: Siemens, 2000: 6

2.11 Dual-Tone Multi-Frequency (DTMF)

Dual-Tone Multi-Frequency (DTMF) merupakan sebuah *tone* yang terbentuk dari dua lapisan frekuensi. Masing-masing merupakan frekuensi pilihan yang mudah

untuk desain filter dan mudah untuk mengirimkan *tone-tone* pada sebuah *line* telepon. *DTMF* didesain untuk mengirimkan kode-kode yang menggunakan mikrofon. Masing-masing beep (atau digit yang ditekan pada telepon) diubah menjadi dua frekuensi secara bersama-sama, yang ditumpangkan pada amplitudo.

Keypad DTMF berbentuk matrik 4x4, dengan masing-masing baris menyatakan frekuensi rendah, dan masing-masing kolom menyatakan frekuensi tinggi. Dengan menekan sebuah tombol, misalnya tombol '1' akan mengirimkan sebuah sinyal *sinusoida* dari kedua frekuensi 697 Hz dan 1209 Hz. Kedua sinyal merupakan alasan mengapa mereka disebut multifrekuensi. Sinyal ini kemudian di-dekode-kan oleh pensaklaran untuk mengetahui tombol mana yang ditekan. Gambar 2.9 menunjukkan frekuensi *keypad DTMF*.

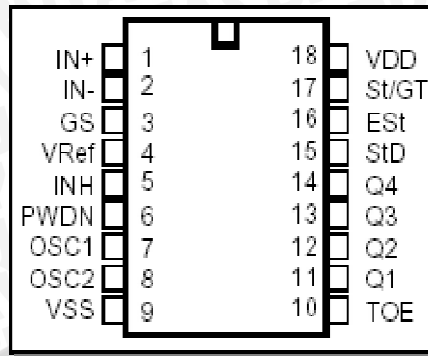
	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Gambar 2.9 Frekuensi *Keypad DTMF*

Sumber: Charrett, 1999: 34

2.12 DTMF Decoder IC MT8870

Decoder/detector DTMF adalah rangkaian yang dapat menerima dan merespons kode-kode sinyal *DTMF* ke dalam bentuk sinyal yang diinginkan. Salah satunya jenis *decoder DTMF* dalam bentuk IC 18 pin adalah IC MT8870. Fungsinya adalah untuk mendeteksi atau menterjemahkan sinyal suara menjadi kode biner 4 bit. Gambar 2.10 menunjukkan konfigurasi pin dari *decoder DTMF* IC MT8870.

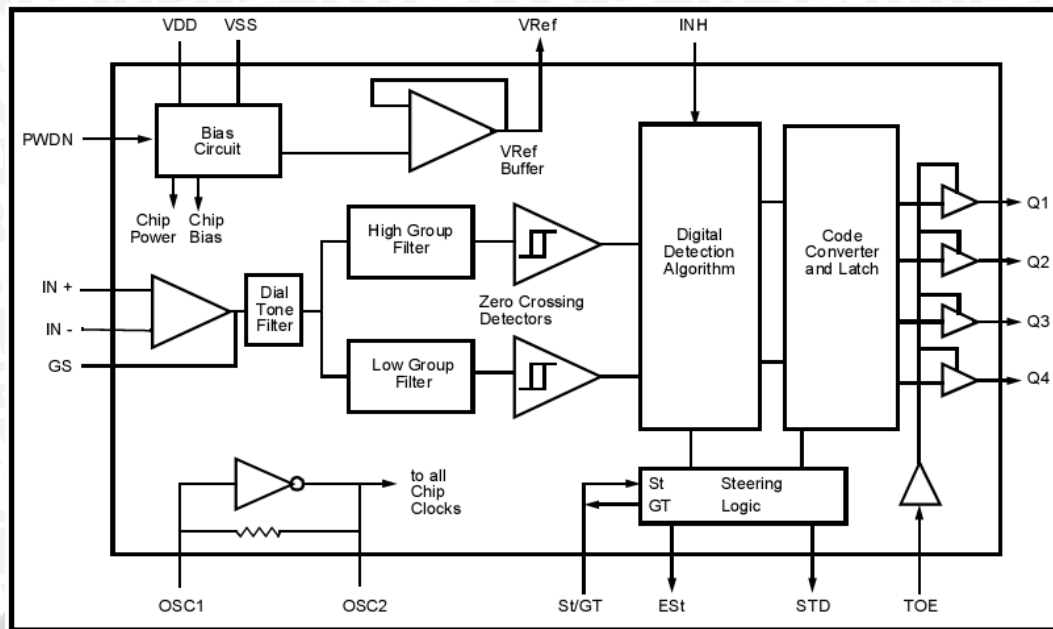
Gambar 2.10 Konfigurasi Pin *Decoder DTMF* IC MT8870

Sumber: Mitel, 1995: 2

Fungsi pin-pin secara keseluruhan dari *decoder DTMF* IC MT8870 adalah sebagai berikut:

- a. IN+ = *Non inverting input op-amp*
- b. IN- = *Inverting input op-amp*
- c. GS = Menghubungkan keluaran penguat differensial dengan resistor umpan balik.
- d. Vref = *Voltage referensi* (keluaran), besarnya $V_{DD}/2$.
- e. INH = Menghasilkan data *DTMF* A, B, C, D
- f. PWDN = *Power down* (masukan), aktif *high*
- g. OSC 1 = *Clock Input*
- h. OSC 2 = *Clock output*, frekuensi keluaran 3.579545 MHz.
- i. Vss = *Ground*, 0 V.
- j. TOE = *Tristate Output Enable*, jika logika *high*, data Q1-Q4 dikeluarkan.
- k. Q1-Q4 = Jika TOE aktif, mengeluarkan kode *DTMF* yang diterima (Tabel 2.4)
Jika TOE logika *low*, data keluaran adalah *high impedance*.
- l. StD = Jika pada pin menghasilkan logika tinggi, maka data keluaran valid.
- m. Est = Saat nilai *tone* masukan benar, maka keluaran pin ini logika tinggi.
- n. St/GT = *Steering/Guard Time*, menentukan kecepatan *tone*.
- o. VDD = Supply tegangan, +5V.

Diagram Blok Fungsi *Decoder DTMF* IC MT8870 ditunjukkan dalam Gambar 2.11 berikut ini.



Gambar 2.11 Diagram Blok Fungsi Decoder DTMF IC MT8870

Sumber: Mitel, 1995: 1

MT8870 merupakan IC tunggal penerima DTMF dengan beberapa bagian bagian didalamnya yang terdiri dari *bandsplit* filter dan *decoder* digital. Bandsplit filter berfungsi untuk memisahkan *tone* DTMF kembali kedalam frekuensi tinggi dan frekuensi rendah, sedangkan Decoder digital memiliki fungsi sebagai bagian yang menverifikasi frekuensi dan durasi kedua *tone* yang telah dipisahkan tersebut dan meneruskannya menjadi 4-bit kode keluaran.

Saat sinyal *input* DTMF masuk, sinyal akan di *buffer* oleh masukan op-amp untuk menyesuaikan nilai penguatan tegangan dan memilih konfigurasi masukan. Kemudian sinyal *input* melalui filter aktif *low pass* dan *dial tone* filter 350Hz/440Hz untuk dipisah kedalam komponen frekuensi *low* dan *high* oleh dua kapasitor-saklar orde enam dan *pass filter*. Masing-masing komponen *tone* kemudian disaring oleh filter keluaran dan diolah menjadi sinyal kotak oleh komparator dengan batas yang kuat.

Dua sinyal kotak yang dihasilkan dikirim ke rangkaian pendeteksi digital untuk didapatkan algoritma hitungan dan periode rata-ratanya. Saat *decoder* melakukan pendeteksian sinyal, flag "Early Steering" akan berlogika naik (0 ke 1). Jika tidak ada sinyal yang dideteksi, Est akan berlogika turun (1 ke 0). Kode yang didapat dikirim ke bagian konverter untuk dikeluarkan pada pin-pin *output* Q1-Q4. Referensi clock yang untuk dapat membaca *tone* DTMF standar dan akurat diperoleh dari kristal *external* 3.579545 MHz.

Timing diagram yang ditunjukkan dalam Gambar 2.12 menggambarkan sinyal kotak dalam setiap waktu proses pendeteksian digital dari tone DTMF. Sinyal frekuensi akan diterima sebagai nilai valid pada masing-masing group tone jika keluaran *Early Steering* (EST) adalah high. Saat perubahan nilai Est dari *low* ke *high*, nilai Vc (pada rangkaian Gambar 4.11) akan meningkat seiring dengan pengosongan Capacitor rangkaian Gambar 4.11. Saat Vc mencapai nilai tegangan treshold (V_{Tst}), Pin *Steering input/guard time output* (St/GT pada Gambar 2.14) berfungsi untuk mengetahui nilai Vc dalam mencapai nilai tegangan V_{Tst} . Saat nilai V_{Tst} sudah tercapai, St/GT akan mengeluarkan tegangan yang menjadikan Vc bernilai Tegangan supply 5 volt (V_{DD}). Lalu pin gerbang keluaran 4 bit akan terbuka dan *output* dapat dikeluarkan.

Tabel 2.6 Tabel Fungsi Decoder DTMF IC MT8870

Digit	TOE	INH	Est	Q ₄	Q ₃	Q ₂	Q ₁
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
A	H	H	L	tidak terdeteksi, kode output yang dihasilkan akan sama dengan keluaran sebelumnya.			
B	H	H	L				
C	H	H	L				
D	H	H	L				

L = Logika Low

H = Logika High

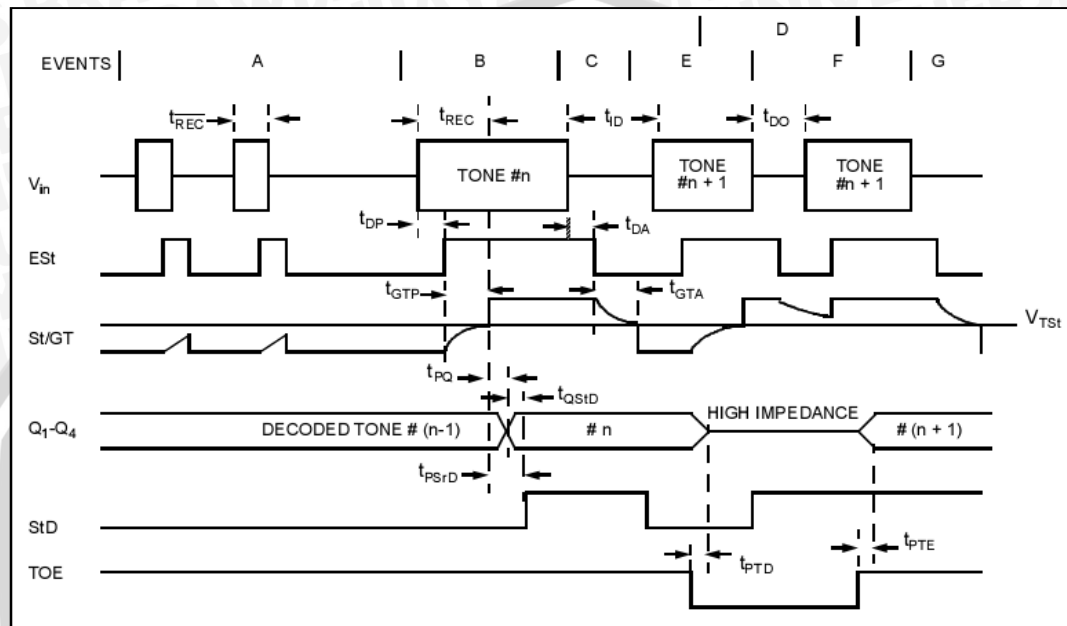
Z = High Impedance

X = Don't Care

Sumber: Mitel, 1995: 4

Besarnya waktu deteksi (saat logika Est naik dari 0 ke 1), t_{GTP} , ditentukan oleh saluran rangkaian RC eksternal. Saat kedua *tone* dapat dipisahkan, sinyal *DTMF*

kemudian akan didekodekan dan data hasilnya di-latch ke register *output*. Keluaran *Delayed Steering* (StD) diberi logika *high* dan menunjukkan bahwa data baru telah siap. Waktu yang diperlukan untuk menerima sinyal *DTMF* valid, t_{REC} , sama dengan jumlah dari t_{DP} dan t_{GTP} . Tabel fungsi *decoder DTMF* ditunjukkan dalam Tabel 2.6.



Gambar 2.12 Diagram Waktu Penerima *DTMF* IC MT8870

Sumber: Mitel, 1995: 4

Keterangan Gambar 2.12:

V_{IN} : Sinyal *DTMF* input

ESt : *Early Steering Output*. Sinyal trigger logika *high* saat IC mendapat *input tone* yang memiliki durasi lebih besar dari t_{REC} .

St/GT : *Steering input/guard time output*. Sinyal yang mengindikasikan waktu deteksi kehadiran *tone input*, pen-decode-an *tone*, dan deteksi berakhirnya *tone*. Saat nilainya lebih tinggi dari V_{TSI} , *tone* yang telah di-decode akan dikirim ke register *output*.

$Q1-Q4$: 4 bit logika *output* dari *tone* yang telah di-decode.

StD : *Delayed Steering Output*. Mengindikasikan bahwa frekuensi valid telah dideteksi dan diproses menjadi 4 *output* $Q1-Q4$ dalam rentang waktu yang dibutuhkan.

TOE : *Tone Output Enable (input)*. Sebuah proses penggeser $Q1-Q4$ menjadi kondisi kosong ke tingkat impedansi High.

- t_{REC} : Durasi minimum sinyal *DTMF* yang dibutuhkan agar pengenalan sinyal *tone* bersifat valid.
- t_{ID} : Waktu minimum antar *input tone-tone DTMF* yang valid.
- t_{DO} : Waktu jatuh/putus maksimal agar *DTMF* sinyal valid.
- t_{DP} : Delay waktu untuk mendeteksi kedatangan sinyal *DTMF* yang valid.
- t_{DA} : Delay waktu untuk mendeteksi akhir sinyal *DTMF* yang valid.
- t_{GTP} : Waktu durasi proses awal decoding sebuah *tone*.
- t_{GTA} : Waktu durasi proses pengakhiran decoding sebuah *tone*.

Proses-proses yang dapat terjadi di dalam IC MT8870 (Gambar 2.12) adalah sebagai berikut

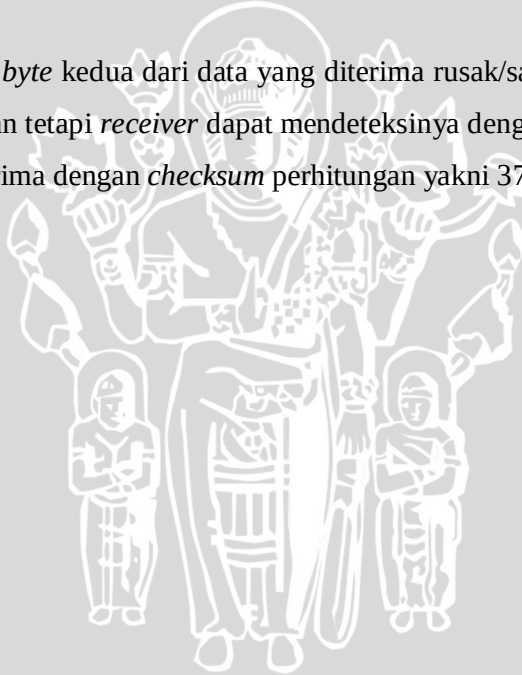
- Pada rentang waktu A. Dua buah sinyal *tone input* terdeteksi. Karena durasi masing-masing *tone* tidak bernilai $\geq t_{REC}$, *tone* dinyatakan tidak valid sehingga *output* tidak dihasilkan.
- Pada rentang waktu B. Sebuah *tone* bernama #n terdeteksi sebagai *input*, Karena durasi *tone* bernilai $\geq t_{REC}$, *tone* dinyatakan valid, *tone* dapat di-*decode* dan diumpankan ke saluran *output*.
- Pada rentang waktu C. Sinyal akhir *tone* bernama #n terdeteksi, durasi waktu kosong valid karena bernilai $\geq t_{ID}$. Pada kondisi ini, kode *tone* #n masih dikeluarkan di saluran *output* Q1-Q4 dan menunggu terdeteksi sinyal *input tone* baru.
- Pada rentang waktu D. Nilai saluran keluaran Q1-Q4 adalah level impedansi high yang berarti tidak menghasilkan kode untuk *tone* tertentu.
- Pada rentang waktu E. Sebuah *tone* lain bernama #n+1 terdeteksi sebagai *input*, durasi *tone* valid karena nilainya $\geq t_{REC}$, *tone* dapat di-*decode* dan diumpankan ke saluran *output*.
- Pada rentang waktu F. Terjadi sinyal putus pada saluran *input*, tetapi masih dapat diterima/dideteksi karena nilainya $\leq t_{DO}$, kode *tone* #n+1 dikeluarkan ke saluran *output*.
- Pada rentang waktu G. Akhir dari *tone* #n+1 terdeteksi, durasi waktu *tone* kosong valid, kode *tone* #n masih dikeluarkan di saluran *output* Q1-Q4 dan menunggu terdeteksi sinyal *input tone* baru.

2.13 Metode Deteksi Kesalahan *Checksum*

Metode deteksi kesalahan *checksum* ini dilakukan dengan cara *transmitter* data memilih suatu nilai (disebut sebagai *checksum*) yang kemudian mengkombinasikannya dengan data yang dikirim menggunakan algoritma tertentu. Dengan algoritma yang sama pula, maka *receiver* melakukan perhitungan terhadap data yang diterima untuk dibandingkan dengan *checksum* dan pada akhirnya dapat diketahui apakah data yang diterima benar atau tidak. Sebagai contoh deteksi kesalahan (*checksum*) sederhana adalah sebagai berikut:

Data	: 6 23 4
Data dengan checksum	: 6 23 4 33
Data setelah transmisi	: 6 27 4 33

Pada data diatas, *byte* kedua dari data yang diterima rusak/salah karena nilainya berubah menjadi 27. Akan tetapi *receiver* dapat mendeteksinya dengan membandingkan *checksum* (33) yang diterima dengan *checksum* perhitungan yakni 37 ($6+27+4$).



BAB III

METODOLOGI PENELITIAN

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasiikan alat yang dirancang adalah perancangan alat, pembuatan alat dan pengujian alat.

3.1 Perancangan Alat

Perancangan alat *Remote Telemetry Unit* (RTU) pada Sistem Pemantau Cuaca Menggunakan *Dial up Handphone* Sebagai Media Transfer Data meliputi tahapan-tahapan sebagai berikut:

- 1) Penentuan spesifikasi alat.
- 2) Pembuatan diagram blok sistem keseluruhan.
- 3) Perancangan perangkat keras masing-masing blok yang meliputi perencanaan dan pembuatan rangkaian dari masing-masing blok.
- 4) Menggabungkan beberapa blok menjadi keseluruhan sistem yang direncanakan.
- 5) Perancangan perangkat lunak mikrokontroler untuk mengendalikan sistem secara keseluruhan.

3.2 Pembuatan Alat

Pembuatan alat *Remote Telemetry Unit* (RTU) pada Sistem Pemantau Cuaca Menggunakan *Dial up Handphone* Sebagai Media Transfer Data meliputi tahapan-tahapan sebagai berikut:

- 1) Pembuatan mekanik sistem berdasarkan pada perancangan.
- 2) Pembuatan perangkat keras sistem dengan menggunakan komponen elektronika yang telah direncanakan.
- 3) Pembuatan perangkat lunak mikrokontroler sesuai dengan diagram alir yang telah direncanakan.

3.3 Pengujian Alat

Untuk mengetahui kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian alat. Pengujian dilakukan pada masing-masing bagian dan

kemudian secara keseluruhan sistem. Secara garis besar pengujian yang dilakukan adalah sebagai berikut.

1) Pengujian Modul LCD 16x2

Pengujian dilakukan dengan menghubungkan LCD dan mikrokontroler yang sudah berisi perangkat lunak untuk menampilkan tulisan tertentu. Pengujian modul LCD ini bertujuan untuk mengetahui keberhasilan LCD menampilkan tulisan sesuai dengan perangkat lunak yang terdapat dalam mikrokontroler.

2) Pengujian Sensor

Pengujian sensor dilakukan dengan cara menyimulasikan sensor dan hasil pemodelan sensor. Pengujian ini bertujuan untuk memastikan sensor dan hasil pemodelan sensor dapat bekerja sesuai dengan perancangan dan memberikan analisis terhadap hasil pengujian.

3) Pengujian pengiriman data secara serial

Pengujian ini dilakukan dengan cara mengirimkan data hasil pengolahan yang dilakukan oleh mikrokontroler ke *hyperterminal* komputer. Pengujian ini bertujuan untuk mengetahui apakah data yang dikirimkan mikrokontroler sudah sesuai dengan apa yang direncanakan dan memberikan analisis terhadap hasil pengujian.

4) Pengujian antarmuka *handphone* dengan mikrokontroler

Pengujian antarmuka *handphone* dengan mikrokontroler dilakukan dengan menghubungkan *handphone* dengan mikrokontroler yang sudah berisi perangkat lunak untuk mengirim perintah-perintah ke *handphone* dan modul LCD. Pengujian ini bertujuan untuk menganalisis dan mengetahui keberhasilan mikrokontroler mengirim perintah-perintah ke *handphone* dan menerima respons dari *handphone* serta memberikan analisis terhadap hasil pengujian.

5) Pengujian penerjemahan sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara *handphone*

Pengujian ini dilakukan dengan cara melewatkan sinyal DTMF yang dihasilkan oleh *handphone* pada kanal suara yang terbentuk ketika terjadi panggilan kemudian menerjemahkan sinyal DTMF tersebut menggunakan rangkaian DTMF *decoder* dan melihat nilai biner hasil penerjemahan sinyal DTMF tersebut. Pengujian ini bertujuan untuk menganalisis dan mengetahui apakah sinyal DTMF yang dibangkitkan *handphone* pada RTU dan dikirimkan ke *handphone* yang ada di *Base Station* dapat diterjemahkan sesuai dengan perancangan.

6) Pengujian sistem secara keseluruhan

Pengujian ini dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat dan memberikan analisis terhadap kinerja alat.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Bab ini akan menjelaskan tentang perancangan sekaligus pembuatan alat. Alat dirancang untuk dapat mengetahui nilai parameter cuaca, memrosesnya dan mengirimkannya ke *base station* melalui kanal suara pada *handphone* yang didahului proses *dial up*.

4.1 Penentuan Spesifikasi Alat

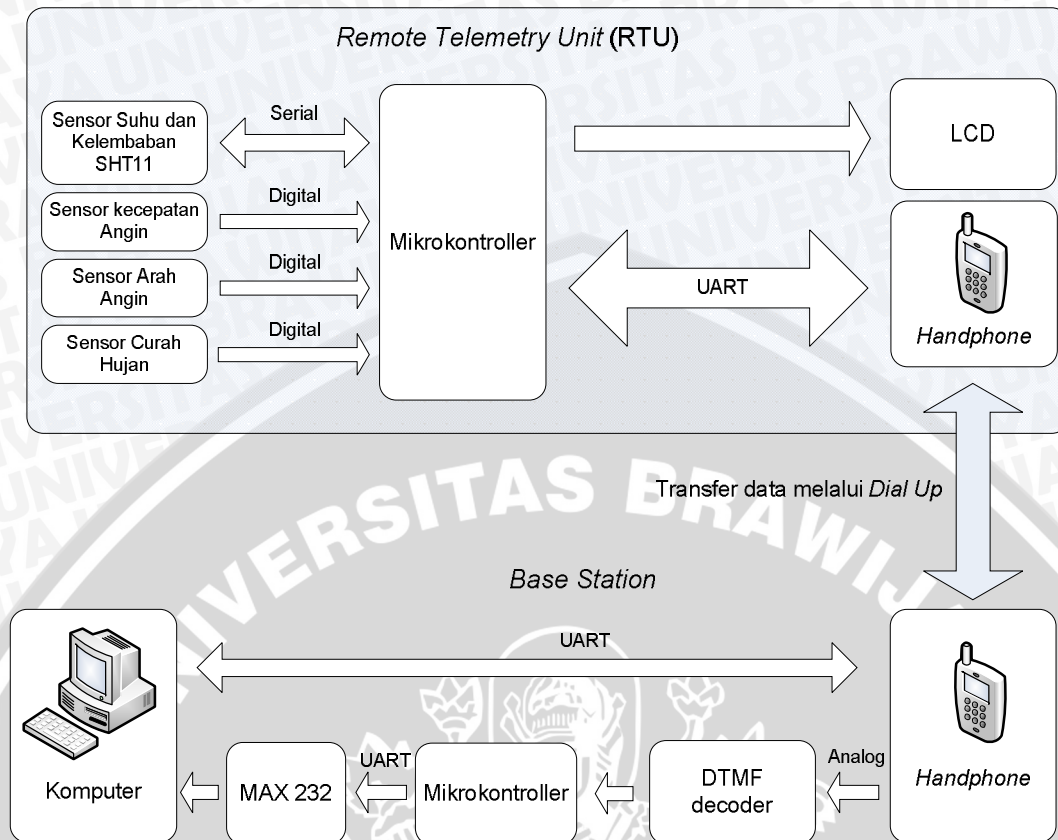
Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

- 1) Jalur transmisi data yang digunakan adalah kanal suara pada *handphone* yang didahului proses *dial up*.
- 2) Suhu udara yang dapat diukur adalah 0 sampai 50°C dengan resolusi 1°C.
- 3) Kelembaban udara yang dapat diukur adalah kelembaban relatif 0% sampai 100% RH dengan resolusi 1% RH.
- 4) Arah angin yang dapat dideteksi ada delapan arah: Utara (U), Timur Laut (TL), Timur (T), Tenggara (Tg), Selatan (S), Barat Daya (BD), Barat (B) dan Barat Laut (BL).
- 5) Kecepatan angin yang dapat diukur adalah 0 sampai 99 km/jam dengan resolusi 1 km/jam.
- 6) Curah hujan yang dapat diukur adalah 0 sampai 300 mm per hari dengan resolusi 1 mm.
- 7) Sistem menggunakan catu daya DC 5-12 volt.

4.2 Diagram Blok Sistem

Secara garis besar, diagram blok perancangan *hardware* sistem secara keseluruhan ditunjukkan dalam Gambar 7.1.

Pada Skripsi ini bagian yang dirancang adalah RTU yang merupakan suatu peralatan yang berada di tempat yang sama dengan besaran yang diamati, dan bertugas mengirimkan data hasil pengamatan ke *Base Station*. Dalam hal ini RTU terdiri atas tiga bagian utama yaitu sensor-sensor parameter cuaca, mikrokontroler dan modul GSM (*handphone*).



Gambar 4.1 Diagram blok sistem secara keseluruhan.

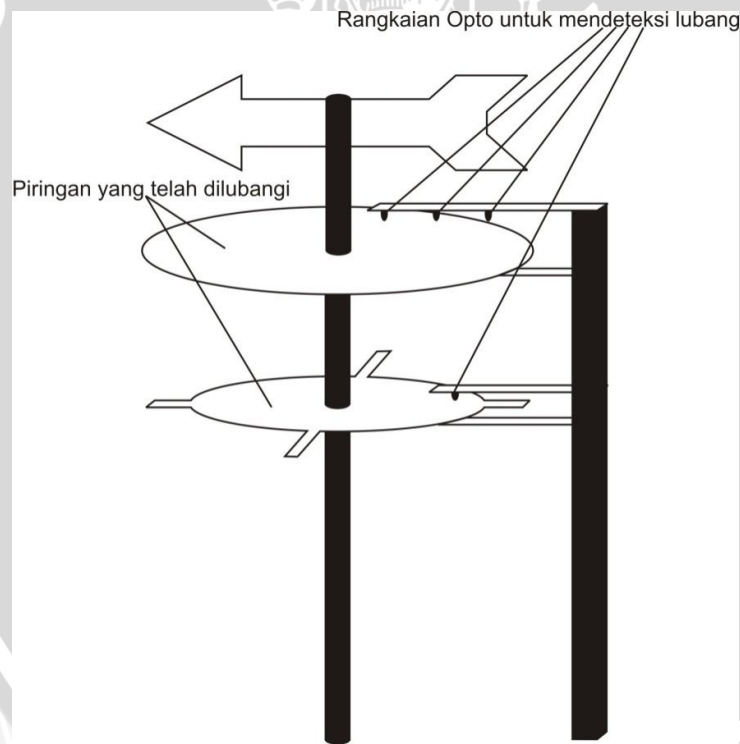
Sensor-sensor parameter cuaca yang terdiri dari empat sensor (sensor suhu dan kelembaban, sensor kecepatan angin, sensor arah angin dan sensor curah hujan) berfungsi mengubah besaran fisis cuaca menjadi besaran elektrik. Besaran-besaran elektrik ini kemudian masuk ke mikrokontroler yang berfungsi sebagai pusat pengolahan dan akuisisi data. Data yang masuk ke mikrokontroler tersebut kemudian diolah sehingga diperoleh data jadi dan format data yang siap ditransmisikan melalui komunikasi serial. Agar dapat ditransmisikan melalui kanal suara *handphone*, maka data yang akan dikirim tersebut harus diubah menjadi analog terlebih dahulu. Dalam hal ini data diubah menjadi sinyal-sinyal DTMF. Sinyal-sinyal DTMF ini dapat langsung dibangkitkan oleh *handphone* yang terhubung dengan mikrokontroler dengan cara memberikan perintah-perintah tertentu pada *handphone*. Sinyal-sinyal DTMF yang dibangkitkan *handphone* ini secara langsung akan masuk ke kanal suara saat terjadi panggilan. Sehingga dengan demikian sinyal-sinyal DTMF yang dibangkitkan oleh *handphone* yang ada di RTU yang merupakan representasi dari data-data digital akan diterima oleh *handphone* yang ada di Base Station. Sinyal-sinyal DTMF yang diterima

handphone yang ada di *Base Station* ini kemudian didekodekan menjadi data-data digital kembali untuk kemudian diolah dan disimpan dalam database.

Proses transmisi data terjadi setelah *base station* melakukan *dial up* pada nomor *handphone* yang terhubung ke mikrokontroler. *Handphone* akan memberikan sinyal ke mikrokontroler bahwa terjadi panggilan, kemudian mikrokontroler akan memberi perintah kepada *handphone* untuk menerima panggilan. Setelah kanal suara terbentuk maka mikrokontroler akan mengirimkan perintah ke *handphone* untuk membangkitkan sinyal-sinyal DTMF sesuai dengan data yang akan ditransmisikan.

4.3 Perancangan Mekanik

Perancangan mekanik dilakukan untuk merancang pemodelan sensor arah dan kecepatan angin. Secara umum rancangan mekanik pemodelan sensor ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Rancangan mekanik pemodelan sensor kecepatan dan arah angin

Dalam Gambar 4.2 terlihat bahwa ada dua buah piringan yang digunakan. Piringan yang terletak di atas adalah piringan yang digunakan untuk mendeteksi arah angin, sedangkan piringan yang di bawah adalah piringan yang digunakan untuk

mendeteksi kecepatan angin. Selain itu terlihat di masing-masing piringan ada rangkaian opto untuk mendeteksi lubang yang ada di masing-masing piringan.

Piringan yang digunakan untuk mendeteksi kecepatan angin berbentuk sebuah lingkaran yang terbagi menjadi tiga daerah. Masing-masing daerah, memiliki daerah yang berlubang dan tidak. Kombinasi dari ketiga daerah tersebut menghasilkan delapan (2^3) kombinasi daerah yang berlubang dan tidak. Setiap kombinasi terdiri dari tiga daerah yang mewakili tiga bit data. Daerah yang berlubang mewakili logika 0, sedangkan daerah yang tidak berlubang mewakili logika 1. Dengan begitu piringan ini dapat digunakan untuk mendeteksi delapan mata angin. Bentuk piringan ini ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Piringan yang digunakan untuk mendeteksi arah angin

Piringan dalam Gambar 4.3 menggunakan tiga buah rangkaian opto untuk mendeteksi arah angin. Sehingga jumlah kombinasi yang bias digunakan dapat dihitung menggunakan persamaan berikut:

$$\text{kombinasi} = 2^n \quad (4-1)$$

$$\text{kombinasi} = 2^3 = 8$$

Dari perhitungan diperoleh delapan kombinasi, dimana setiap kombinasi mewakili satu arah angin. Kombinasi logika untuk masing-masing arah angin dapat dilihat dalam Tabel 4.1.

Tabel 4.1 Kombinasi logika untuk masing-masing arah angin

Kombinasi Logika	Arah Angin
000	Utara (U)
001	Barat Laut (BL)
011	Barat (B)
010	Barat Daya (BD)
110	Selatan (S)
111	Tenggara (Tg)
101	Timur (T)
100	Timu Laut (TL)

Pirigan kedua adalah piringan yang digunakan untuk mendeteksi kecepatan angin. Piringan ini hanya memiliki satu daerah. Piringan ini mempunyai empat buah lubang, sehingga untuk satu kali putaran akan dideteksi empat kali logika 0. Sedangkan untuk mengkonversi menjadi nilai kecepatan angin digunakan Persamaan 2-1 dan Persamaan 2-2.

Dalam perancangan sensor kecepatan angin ini resolusi yang digunakan adalah 1 km/jam dengan waktu sampling untuk kecepatan sesaat adalah 1 detik. Dalam setiap 1 detik sistem akan mengecek jumlah pulsa yang terbentuk. Jumlah pulsa yang terbentuk tersebut merupakan nilai kecepatan sesaat. Kondisi ini dapat dituliskan dalam persamaan berikut:

$$S = n \text{ km/jam} \quad (4-2)$$

Dimana S adalah kecepatan angin (km/jam), sedangkan n adalah jumlah pulsa yang terbentuk selama 1 detik.

Piringan yang dirancang memiliki empat buah lubang sehingga dalam satu putaran akan terjadi empat buah pulsa. Sehingga Persamaan 4-2 dapat berubah menjadi:

$$S = 4R \text{ km/jam} \quad (4-3)$$

Dimana R adalah jumlah putaran yang terjadi dalam 1 detik.

Apabila satuan dalam Persamaan 4-3 diubah dalam cm/detik maka penurunan rumusnya adalah sebagai berikut:

$$S = 4R \times 100.000 \text{ cm/3600 detik}$$

$$S = 111,111R \text{ cm/detik} \quad (4-4)$$

Dengan mensubstitusikan Persamaan 2-2 dengan Persamaan 4-4 maka akan diperoleh jarak yang ditempuh oleh piringan (D) dengan perhitungan sebagai berikut:

$$S = D/T$$

$$111,111R \text{ cm/detik} = D/T$$

$$T = 1 \text{ detik}$$

$$D = 111,111R \text{ cm} \quad (4-5)$$

Dengan menggunakan Persamaan 2-1 dan Persamaan 4-5, maka akan didapat diameter piringan yang dirancang dengan perhitungan sebagai berikut:

$$D = \pi dR$$

$$111,111R \text{ cm} = \pi dR$$

$$d = 111,111 \text{ cm} / \pi$$

$$d = 35,368 \text{ cm}$$

Pembuatan mekanik pemodelan sensor kecepatan dan arah angin ini dilakukan dengan menyusun semua bagian yang dibutuhkan sesuai perancangan. Hasil pembuatan mekanik pemodelan sensor kecepatan angin, arah angin dan curah hujan ditunjukkan dalam gambar 4.4.



Gambar 4.4 Hasil pembuatan mekanik pemodelan sensor kecepatan angin, arah angin dan curah hujan

4.4 Perancangan Perangkat Keras (*Hardware*)

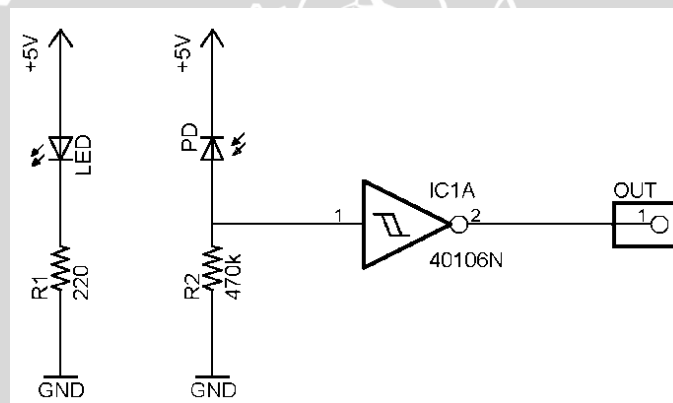
Perancangan perangkat keras terdiri atas perancangan rangkaian *push button* untuk pemodelan *tipping bucket* sebagai sensor curah hujan, rangkaian opto untuk pemodelan sensor kecepatan dan arah angin, rangkaian antarmuka SHT11 dengan mikrokontroler, rangkaian DTMF *decoder* dan rangkaian antarmuka *handphone* dengan mikrokontroler.

4.4.1 Perancangan Rangkaian Opto Untuk Pemodelan Sensor Kecepatan dan Arah Angin

Untuk membuat pemodelan sensor kecepatan dan arah angin digunakan rangkaian opto yang terdiri dari LED dan *photodiode*. LED berfungsi sebagai sumber

cahaya dan *photodiode* berfungsi sebagai penerima cahaya. Dua komponen ini disusun berhadapan dengan sebuah piringan di tengahnya, sehingga cahaya dari LED akan langsung diterima *photodiode* jika tidak ada penghalang di antara dua komponen ini dan juga sebaliknya, jika ada penghalang maka cahaya dari LED tidak akan diterima oleh *photodiode*.

Rangkaian ini akan memberikan logika yang berbeda ketika terdapat penghalang diantara LED dan fotodioda. Pada pemodelan sensor arah angin digunakan digunakan tiga buah LED dan fotodioda yang disusun secara berjejer yang di tengahnya diberi piringan seperti pada Gambar 4.3. Pemodelan ini mampu menunjukkan delapan mata angin. Pada pemodelan sensor kecepatan angin digunakan sepasang LED dan *photodiode* yang akan menghitung banyaknya putaran yang terjadi pada piringan anemometer. Rangkaian opto yang terdiri atas LED dan *photodiode* ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Rangkaian opto yang terdiri atas LED dan *photodiode*

Untuk mendapatkan nilai-nilai resistor yang terpasang pada LED dan *photodiode* dilakukan perhitungan sebagai berikut:

Pada LED diketahui berdasarkan datasheet arus maksimum adalah 20 mA dan tegangan maju pada LED adalah 1,8 V, sehingga perhitungan R_1 adalah sebagai berikut:

$$V_{CC} = V_{LED} + IR_1$$

$$R_1 = \frac{V_{CC} - V_{LED}}{I}$$

$$R_1 = \frac{5 - 1,8}{20 \cdot 10^{-3}}$$

$$R_1 = 160\Omega$$

Nilai 160 ohm adalah nilai R untuk arus maksimum. Karena di pasaran tidak ada R dengan nilai 160 ohm, maka digunakan nilai R yang lebih besar yaitu 220 ohm.

Pada *photodiode* pada saat terkena cahaya dari LED (tidak terhalang piringan) diperoleh I_L (*light current*) secara umum berdasarkan datasheet adalah $10 \mu A$. pada kondisi ini tegangan pada *photodiode* (V_{fd}) sangat kecil sehingga dapat diabaikan dalam perhitungan. Perhitungan R_2 adalah sebagai berikut:

$$V_{CC} = I_L R_2$$

$$R_2 = \frac{V_{CC}}{I_L}$$

$$R_2 = \frac{5}{10 \cdot 10^{-6}}$$

$$R_2 = 500k\Omega$$

Karena nilai 500 kohm tidak ada di pasaran, maka digunakan nilai 470 kohm.

Untuk mengurangi *noise* dan memastikan logika keluaran dari *photodiode* digunakan *trigger smith* (IC HEF40106BP). Keluaran *trigger smith* ini yang akan terhubung ke mikrokontroler.

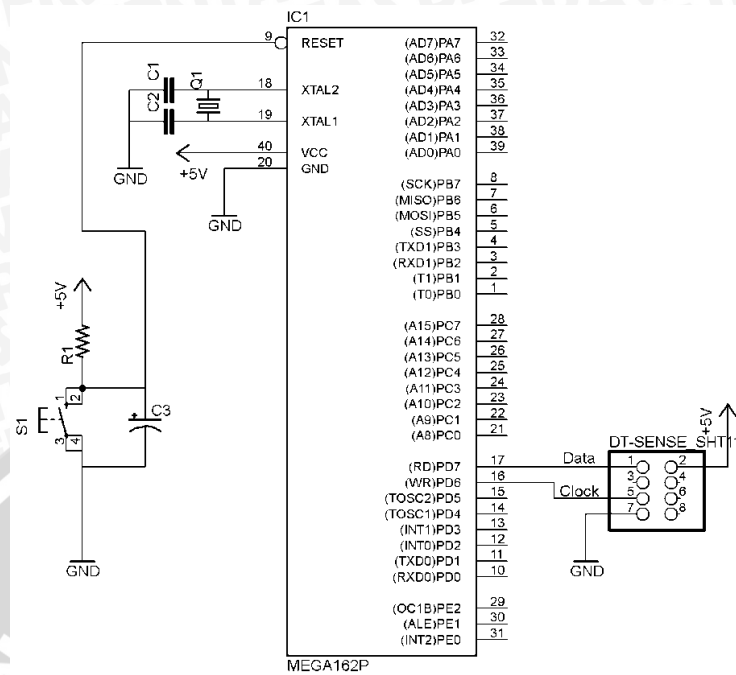
4.4.2 Perancangan Rangkaian Antarmuka SHT11 Dengan Mikrokontroler

Untuk mendeteksi suhu dan kelembaban digunakan SHT11, sebuah sensor keluaran sensirion. Dalam perancangan ini digunakan DT-Sense SHT11, sebuah modul jadi dari sensor SHT11 yang telah dilengkapi rangkaian antarmuka. Bentuk modul ini ditunjukkan dalam Gambar 4.6.

Modul DT-Sense SHT11 ini memiliki delapan pin keluaran yang terdiri atas satu pin data, satu pin *clock*, satu pin V_{CC} , satu pin GND dan empat pin NC (*Not Connect*). Pin modul DT-Sense SHT11 yang terhubung ke mikrokontroler adalah pin data dan *clock*. Rangkaian antarmuka DT-Sense SHT11 dengan mikrokontroler ATmega8 ditunjukkan dalam Gambar 4.7.



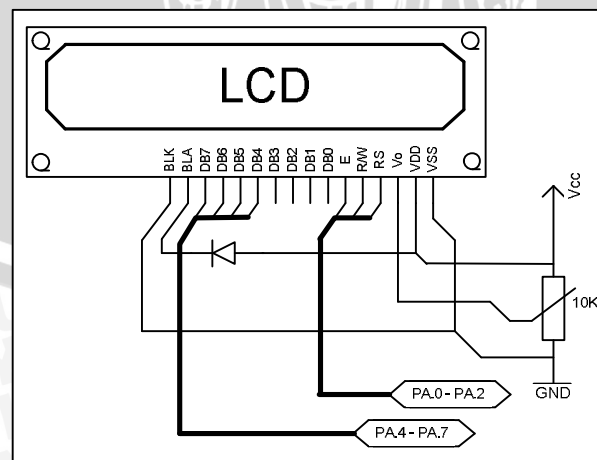
Gambar 4.6 Modul DT-Sense SHT11



Gambar 4.7 Rangkaian antarmuka DT-Sense SHT11 dengan mikrokontroler ATmega162

4.4.3 Perancangan rangkaian antarmuka LCD LMB162ADC

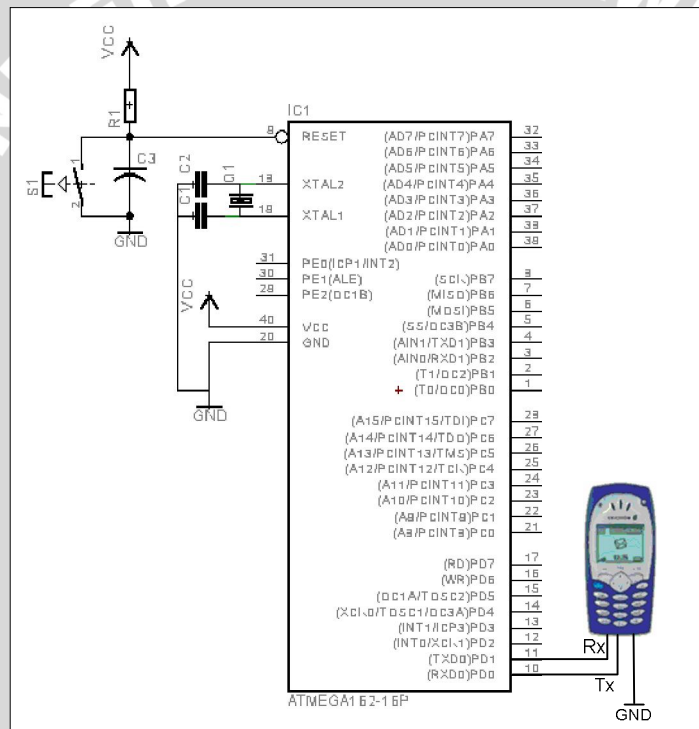
Pada perancangan ini LCD LMB162ADC digunakan untuk menampilkan nilai parameter cuaca hasil pengolahan data oleh mikrokontroler. Pin RS dari LCD terhubung ke PORTA.0 mikrokontroler, pin R/W dari LCD terhubung ke PORTA.1 dan pin *Enable* dari LCD terhubung ke PORTA.2 dari mikrokontroler. Sedangkan pin data dari LCD (PB4 sampai PB7) dihubungkan dengan PORTA nomor 4 sampai nomor 7 dari mikrokontroler. Gambar rangkaian dari LCD LMB162ADC ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Rangkaian LCD LMB162ADC

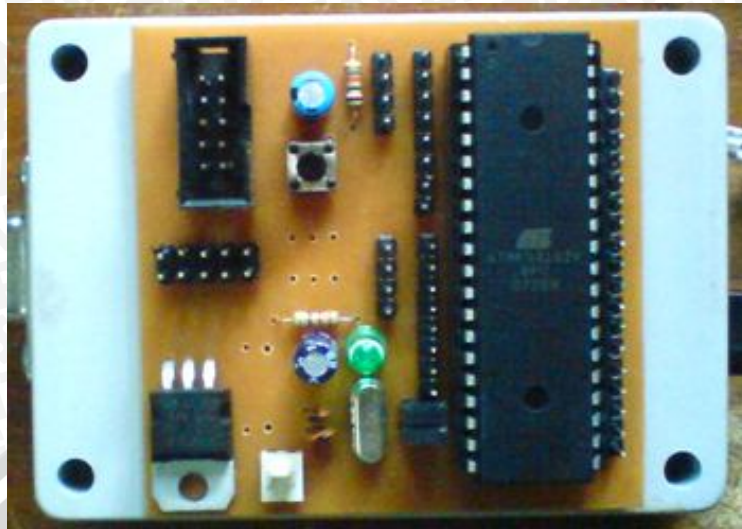
4.4.4 Perancangan Rangkaian Antarmuka *Handphone* Dengan Mikrokontroler

Dalam perancangan ini *handphone* digunakan untuk menerima panggilan dari *Base Station* dan sebagai media transfer data melalui kanal suara yang terbentuk pada saat terjadi panggilan. Untuk menghubungkan *handphone* dengan mikrokontroler ATmega162 dilakukan dengan menghubungkan pin Tx, Rx dan *ground* *handphone* ke pin Rx, Tx dan *ground* mikrokontroler ATmega162. Selain itu untuk dapat berkomunikasi secara serial diperlukan kecepatan data (*baudrate*) sebesar 19200 bps. Rangkaian *interface* *handphone* ke mikrokontroler ATmega162 ditunjukkan dalam Gambar 4.9.



Gambar 4.9 Interface *Handphone* Siemens C45 ke Mikrokontroler ATmega162

Pembuatan rangkaian antarmuka mikrokontroler dengan *handphone*, LCD, SHT11 dan rangkaian *opto* dijadikan satu dalam sebuah minimum sistem ATmega162, yang dilakukan dengan menyusun semua komponen yang dibutuhkan sesuai perancangan. Hasil pembuatan minimum sistem ATmega162 ini ditunjukkan dalam Gambar 4.10



Gambar 4.10 Hasil pembuatan minimum sistem ATMega162

4.4.5 Perancangan Rangkaian DTMF Decoder

Rangkaian *DTMF decoder* menggunakan komponen utama MT8870 sebagai penerima sinyal *DTMF* dari *line-out headset handphone* yang kemudian diubah ke format biner 4 bit. Konfigurasi rangkaian penerima *DTMF* digambarkan dalam Gambar 4.11. Rangkaian MT8870 dilengkapi dengan kristal (3,579545MHz), C_2 dan R_3 yang dipakai untuk menentukan waktu minimal untuk mengenali nada *DTMF* yang diterima, dan rangkaian penguat sinyal *DTMF* yang terdiri dari R_1 , dan R_2 serta kapasitor coupling C_1 .

Rangkaian *DTMF decoder* yang dirancang memiliki spesifikasi sebagai berikut:

- *Supply* tegangan yang dibutuhkan pada IC *DTMF*, $V_{DD} = 5$ volt.
- *Input* berupa sinyal suara *tone DTMF* dengan nilai tegangan puncak-ke-puncak minimum adalah 0,04 volt dan maksimumnya adalah 1,2 volt. Sinyal ini dikirim oleh *handphone transmitter* dan dikeluarkan oleh *line-out headset handphone receiver*.
- *Output* utama berupa 4 buah sinyal digital yang keluar secara paralel saat sebuah *tone input* telah berhasil di-encode.
- *Output* tambahan adalah sinyal StD yang menandakan data *output* utama siap diambil.

Arus *input* yang dibutuhkan rangkaian IC adalah arus AC murni, sehingga untuk menahan arus DC agar tidak masuk ke rangkaian, pada jalur masukan ditambahkan kapasitor coupling (C_1). Sesuai dengan konfigurasi *datasheet*,

Pada bagian penguatan tegangan, dimisalkan nilai resistansi R_1 telah ditentukan sebesar $100\text{ k}\Omega$. Resistor R_2 digunakan sebagai umpan balik untuk menentukan penguatan pada penguat kerja. Karena nilai sinyal masukan sebesar sudah masuk kriteria kerja IC MT8870, maka tidak perlu dilakukan penguatan sinyal masukan.

Nilai penguatan sinyal *input* didapat dari persamaan perbandingan R_1 dan R_2 yang terhubung ke pin *Gain Select* (GS). Nilai-nilai tersebut diperoleh dari:

$$A_v = \frac{R_2}{R_1}$$

Jika nilai telah dimisalkan $R_1 = 100\text{ k}\Omega$ dan penguatan yang diinginkan (A_v) adalah sebesar 1 kali, maka:

$$1 = \frac{R_2}{100\text{ k}\Omega}$$

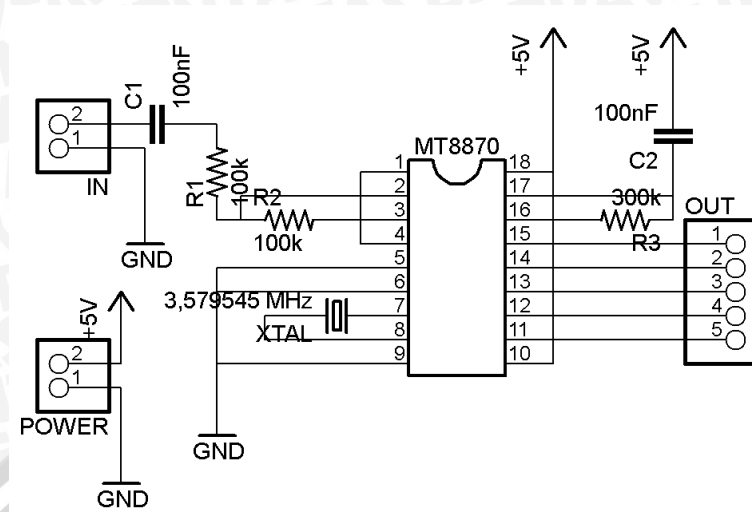
$$R_2 = 100\text{ k}\Omega$$

Kecepatan pemrosesan sinyal pada *DTMF* dari saat kombinasi nada diterima hingga dikeluarkan dalam bentuk biner 4 bit ditentukan oleh resistor R_3 dan C_2 . Pada perancangan rangkaian *DTMF decoder* ini nilai R_3 dan C_2 disesuaikan dengan rangkaian yang telah direkomendasikan dalam *datasheet* yaitu sebagai berikut:

- $R_3 = 300\text{ k}\Omega$
- $C_2 = 0,1\text{ }\mu\text{F}$

Rangkaian *DTMF decoder* secara lengkap ditunjukkan dalam Gambar 4.11.

Pembuatan *DTMF decoder* dilakukan dengan menyusun semua komponen yang dibutuhkan sesuai perancangan. Hasil pembuatan *DTMF decoder* ini ditunjukkan dalam Gambar 4.12.



Gambar 4.11 Konfigurasi rangkaian DTMF decoder



Gambar 4.12 Hasil pembuatan rangkaian DTMF decoder

4.5 Perancangan Format Data

Agar komunikasi dapat berjalan dengan baik dan data yang dikirimkan dapat dikenali oleh penerima maka diperlukan perancangan format data. Format data yang digunakan untuk melakukan komunikasi antara mikrokontroler ATmega162 dengan komputer *Base Station* ditunjukkan dalam Gambar 4.13.

Daerah	Suhu	Kelembaban	Kecepatan angin	Arah angin	Curah hujan	Cek error	Ekor
-	--	--	--	-	--	---	*

Gambar 4.13 Format data komunikasi antara mikrokontroler ATmega162 dengan komputer *Base Station*

Penjelasan dari format data dalam Gambar 4.13 adalah sebagai berikut:

Daerah	: sebuah karakter (1 byte) yang digunakan untuk menandai daerah asal data.
Suhu	: dua buah karakter (2 byte) yang menyatakan nilai suhu.
Kelembaban	: dua buah karakter (2 byte) yang menyatakan nilai kelembaban.
Kecepatan Angin	: dua buah karakter (2 byte) yang menyatakan nilai kecepatan angin.
Arah Angin	: sebuah karakter (1 byte) yang menyatakan arah angin.
Curah Hujan	: dua buah karakter (2 byte) yang menyatakan nilai suhu.
Cek Error	: tiga karakter (3 byte) data yang digunakan untuk mengecek kebenaran data yang dikirim.
Ekor	: sebuah karakter (1 byte, yaitu karakter '*') yang ditempatkan di bagian akhir yang menandakan akhir dari pengiriman data.

4.6 Perancangan Protokol Komunikasi

Agar terjadi komunikasi antara RTU dan *Base Station* diperlukan suatu protokol komunikasi. Protokol komunikasi yang dirancang adalah komunikasi antara mikrokontroler ATmega162 dengan komputer. Protokol antara mikrokontroler ATmega162 dengan komputer adalah sebagai berikut:

1. Mikrokontroler ATmega162 akan mengirim data ke komputer yang terdiri dari kode daerah, nilai parameter cuaca, error cek dan ekor sesuai dengan format data yang telah dirancang.
2. Proses komunikasi dimulai setelah komputer *Base Station* memberikan perintah pada *handphone* yang terhubung ke komputer untuk melakukan *dial up* kepada *handphone* yang berada di RTU.
3. Ketika terjadi panggilan, maka *handphone* yang berada di RTU akan memberikan sinyal kepada mikrokontroler ATmega162 yang menandakan sedang terjadi panggilan.
4. ATmega162 kemudian memberikan perintah kepada *handphone* untuk menerima panggilan tersebut.
5. Setelah panggilan diterima dan kanal suara terbentuk maka ATmega162 mengirimkan data sesuai dengan format data yang telah dirancang melalui kanal suara.

6. Setelah data selesai dikirim maka komputer Base Station memberikan perintah kepada *handphone* untuk menutup panggilan sehingga kanal suara terputus.
7. Komputer di *Base Station* kemudian mengecek data yang diterima. Jika data yang dikirim oleh ATMega162 tidak sama dengan data yang diterima oleh komputer yang ditandai dengan ketidaksesuaian data *cek error*, maka komputer akan memerintahkan *handphone* yang terhubung ke komputer untuk melakukan *dial up* lagi ke *handphone* yang ada di RTU dan proses berulang kembali ke nomer tiga.
8. Jika data yang dikirim oleh ATMega162 sama dengan data yang diterima oleh komputer yang ditandai dengan kesamaan data *cek error*, maka komputer akan memproses data tersebut dan proses komunikasi data berakhir.

4.7 Perancangan Perangkat Lunak Mikrokontroler

RTU terdiri dari sensor-sensor, modulator dan sebuah mikrokontroler yang berfungsi untuk mengakuisisi data dari sensor-sensor dan mengatur komunikasi data. Untuk dapat menjalankan fungsinya dengan baik maka mikrokontroler perlu deprogram terlebih dahulu.

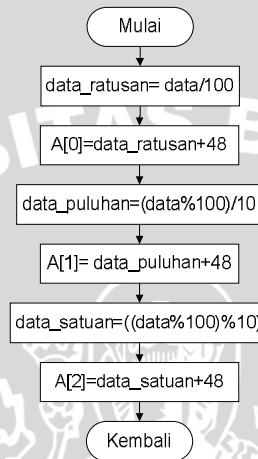
Program dibuat secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu agar memudahkan dalam pembuatan dan pencarian kesalahan serta pengujian program sehingga dapat bekerja dengan baik. Perangkat lunak mikrokontroler dibuat menggunakan *Code Vision AVR* buatan HP *info tech*. Bahasa pemrograman yang digunakan adalah bahasa C.

4.7.1 Perancangan Sub Rutin Mengubah Nilai Integer ke ASCII

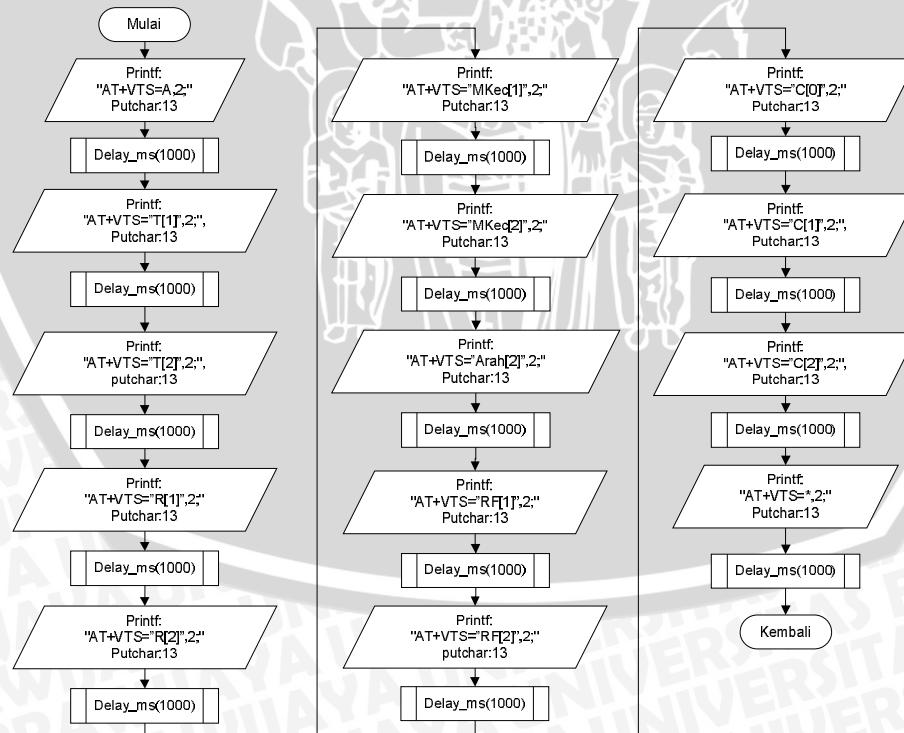
Sub rutin mengubah nilai integer ke ASCII ini berfungsi untuk mengubah tipe data parameter cuaca yang awalnya adalah integer menjadi karakter-karakter yang sesuai dengan ASCII. Pengubahan ini perlu dilakukan karena untuk merepresentasikan data-data parameter cuaca menjadi sinyal-sinyal DTMF, maka mikrokontroler harus mengirimkan perintah-perintah ke *handphone* dalam bentuk karakter-karakter. Dalam sub rutin ini program akan mengambil nilai ratusan dari data yang akan diubah, kemudian mengubahnya menjadi karakter. Selanjutnya program akan mengambil nilai ratusan dan mengubahnya menjadi karakter, dan yang terakhir program akan mengambil data satuan dan mengubahnya menjadi karakter. Diagram alir sub rutin mengubah nilai integer ke ASCII ini ditunjukkan dalam Gambar 4.14.

4.7.2 Perancangan Sub Rutin Mengirim Data

Sub rutin mengirim data ini berfungsi untuk memerintahkan *handphone* membangkitkan sinyal-sinyal DTMF sesuai dengan data-data parameter cuaca hasil pengolahan mikrokontroler. Dalam sub rutin ini program akan mengirimkan perintah “AT+VTS=[data],1” ke *handphone* secara berturut-turut sesuai dengan data parameter cuaca yang ada. Diagram alir sub rutin mengirim data ini ditunjukkan dalam Gambar 4.15.



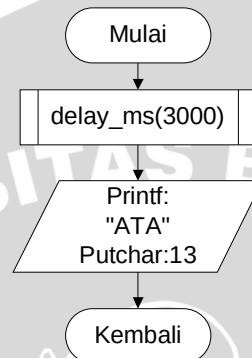
Gambar 4.14 Diagram alir sub rutin mengubah nilai integer ke ASCII



Gambar 4.15 Diagram alir sub rutin mengirim data ini ditunjukkan dalam

4.7.3 Perancangan Sub Rutin Menerima Panggilan

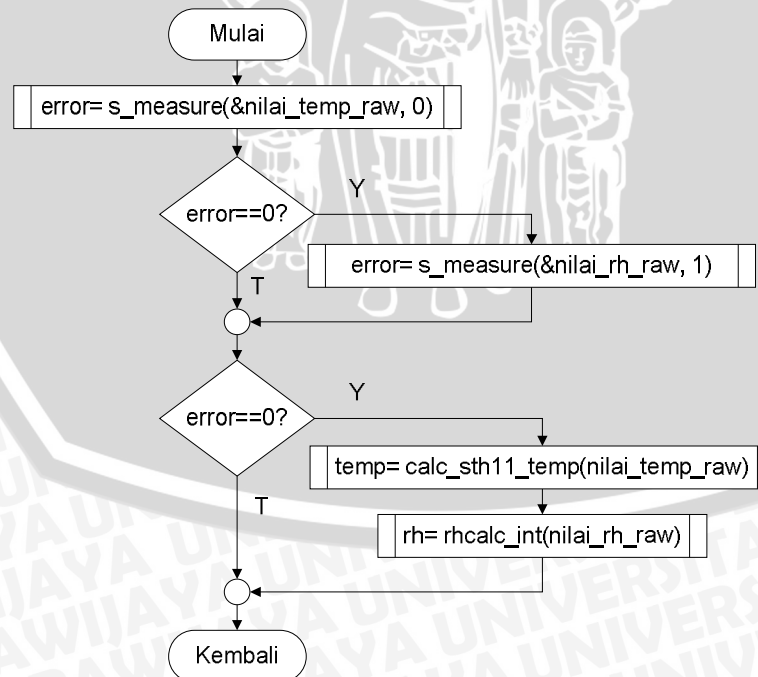
Sub rutin menerima panggilan ini berfungsi untuk memberi perintah ke *handphone* pada saat terjadi panggilan. Penerimaan panggilan ini perlu dilakukan untuk membentuk kanal suara yang menjadi media transmisi data. Dalam sub rutin ini mikrokontroler mengirimkan perintah “ATA” ke *handphone* untuk menerima panggilan. Diagram alir sub rutin menerima panggilan ditunjukkan dalam Gambar 4.16.



Gambar 4.16 Diagram alir sub rutin menerima panggilan

4.7.4 Perancangan Sub Rutin Mengambil Data SHT11

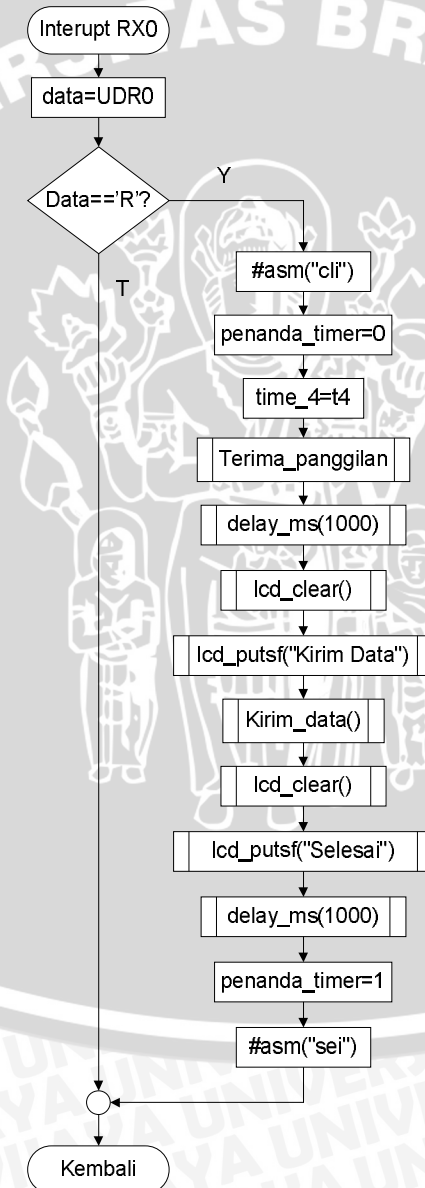
Sub rutin mengambil data SHT11 ini berfungsi untuk mendapatkan data suhu dan kelembaban dari modul SHT11. Diagram alir sub rutin mengambil data SHT11 ini ditunjukkan dalam Gambar 4.17.



Gambar 4.17 Diagram alir sub rutin mengambil data SHT11

4.7.5 Perancangan Sub Rutin *Interrupt* RX

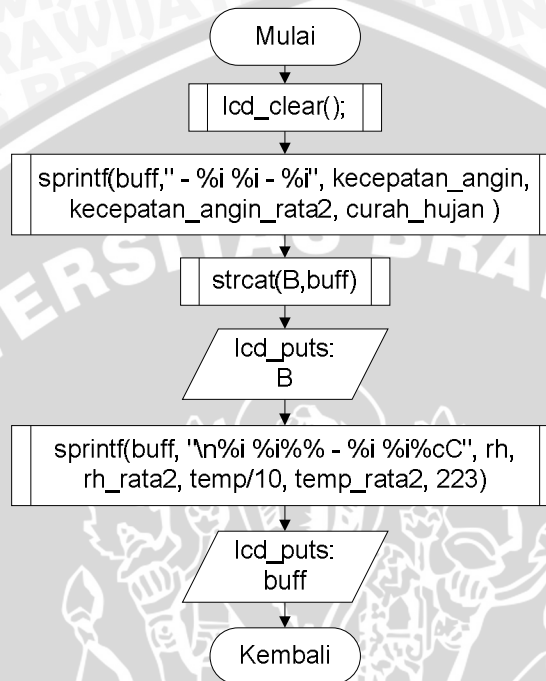
Sub rutin *interrupt* RX ini berfungsi untuk menangani terjadinya panggilan pada *handphone*. Apabila terjadi panggilan, maka *handphone* akan mengirimkan string “RING” kepada mikrokontroler. Apabila mikrokontroler menerima satu karakter dari *handphone*, maka *interrupt* RX akan dijalankan. Dalam subrutin *interrupt* RX ini mikrokontroler akan mengecek terlebih dahulu apakah karakter yang diterima sama dengan ‘R’, yang menandakan terjadinya panggilan. Jika karakter yang diterima adalah ‘R’, maka mikrokontroler akan menerima panggilan dan mengirimkan data. Diagram alir sub rutin *interrupt* RX ditunjukkan dalam Gambar 4.18.



Gambar 4.18 Diagram alir sub rutin *interrupt* RX

4.7.6 Perancangan Sub Rutin Menampilkan Data pada LCD

Sub rutin menampilkan data pada LCD ini berfungsi untuk menampilkan data-data parameter cuaca hasil pengolahan yang dilakukan mikrokontroler ke LCD. Diagram alir sub rutin menampilkan data pada LCD ini ditunjukkan dalam Gambar 4.19.



Gambar 4.19 Diagram alir sub rutin menampilkan data pada LCD

4.7.7 Perancangan Sub Rutin Pengecekan Koneksi *Handphone*

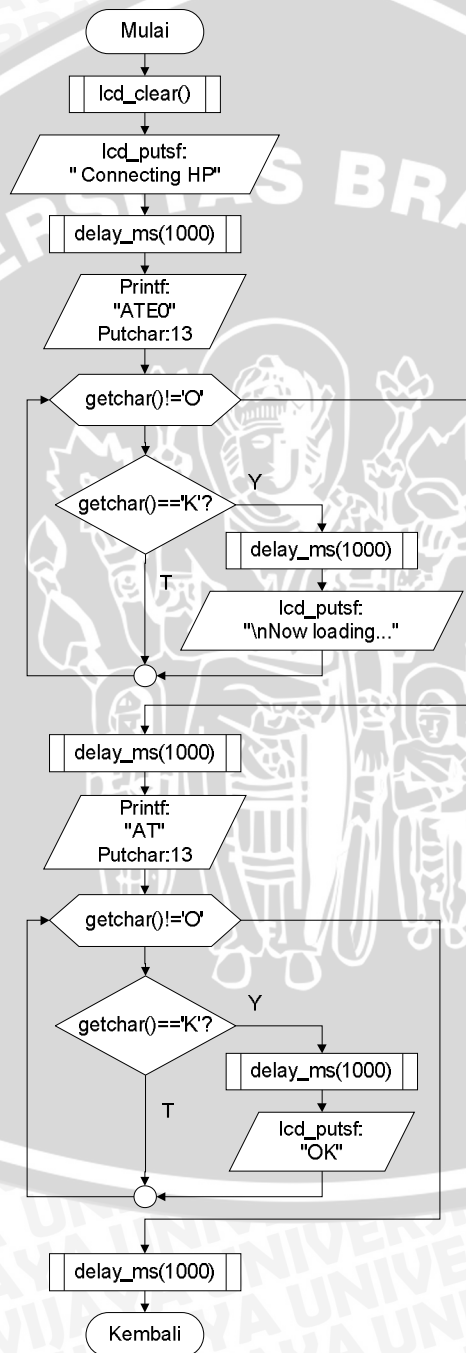
Sub rutin pengecekan koneksi *handphone* ini berfungsi untuk mengecek apakah *handphone* memberikan respons yang sesuai dengan perintah yang diterimanya. Dalam sub rutin ini mikrokontroler mengirimkan perintah “ATE0” dan “AT”, jika respon yang diterima mikrokontroler adalah “OK” maka mikrokontroler akan menampilkan ke LCD bahwa mikrokontroler sudah terkoneksi ke *handphone*. Diagram alir subrutin pengecekan koneksi *handphone* ini ditunjukkan dalam Gambar 4.20

4.7.8 Perancangan Sub Rutin Mengambil Data Arah Angin

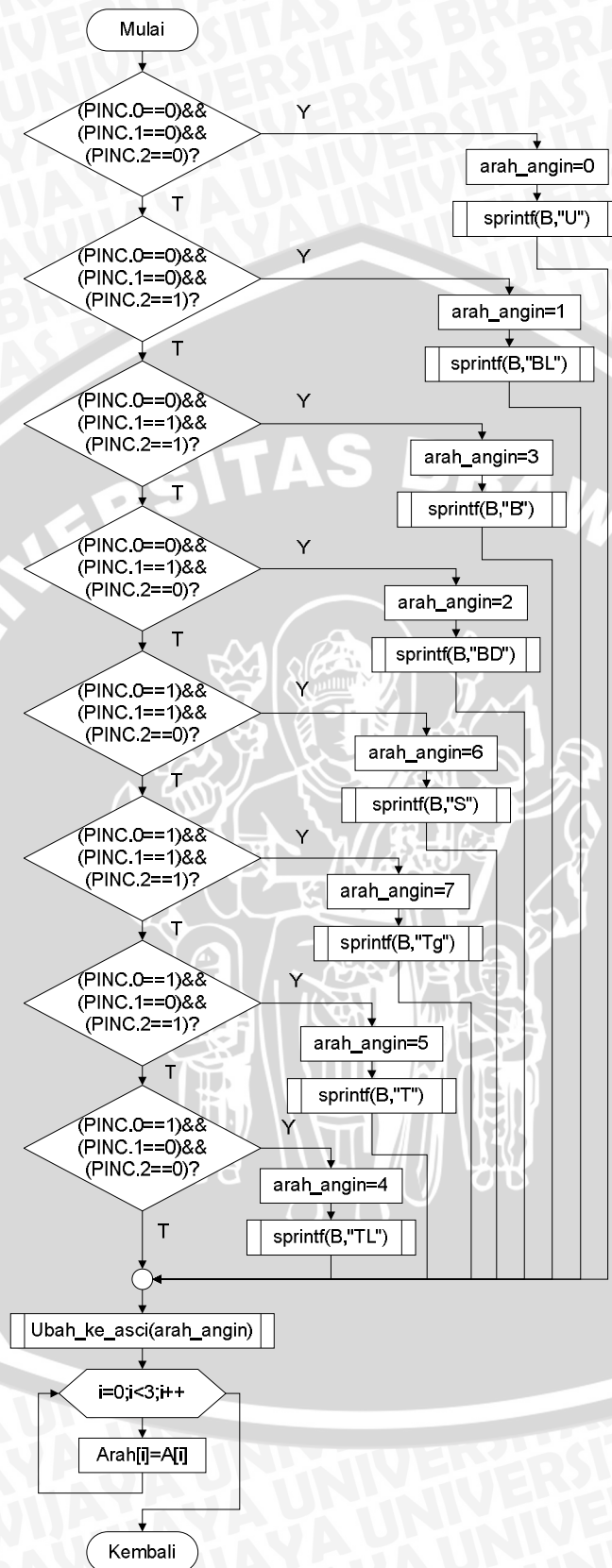
Sub rutin mengambil data arah angin ini berfungsi untuk mengambil data dari pemodelan sensor arah angin. Dalam sub rutin ini mikrokontroler akan mengecek *PIN* mana yang aktif dan menentukan arah angin berdasarkan kondisi *PIN*. Diagram alir subrutin mengambil data arah angin ini ditunjukkan dalam Gambar 4.21.

4.7.9 Perancangan Sub Rutin Pengecekan *Limit Switch*

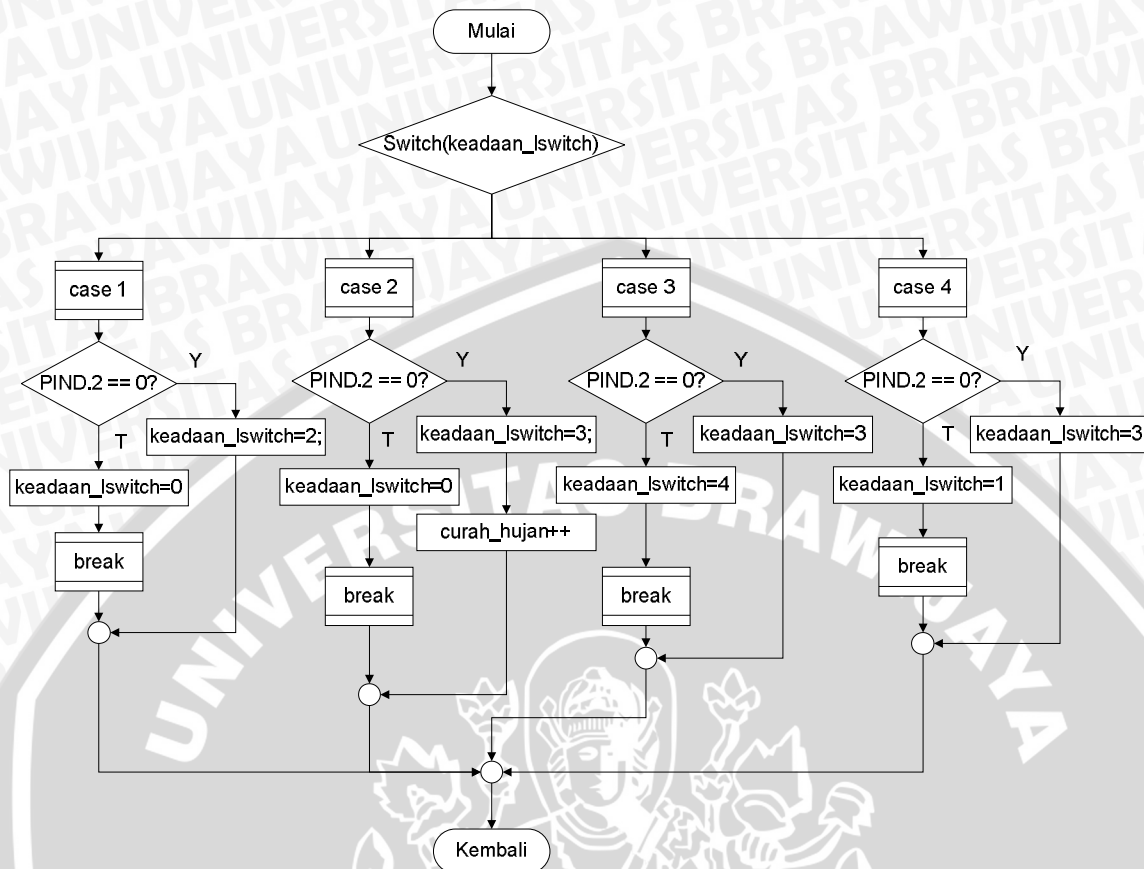
Sub rutin pengecekan *limit switch* ini berfungsi untuk mengecek apakah *limit switch* yang digunakan untuk pemodelan sensor curah hujan ini tertekan atau tidak, untuk melakukan penghitungan jumlah penekanan dan mengatasi bouncing yang ada pada *limit switch*. Diagram alir sub rutin pengecekan *limit switch* ini ditunjukkan dalam Gambar 4.22.



Gambar 4.20 Diagram alir subrutin pengecekan koneksi *handphone*



Gambar 4.21 Diagram alir subrutin mengambil data arah



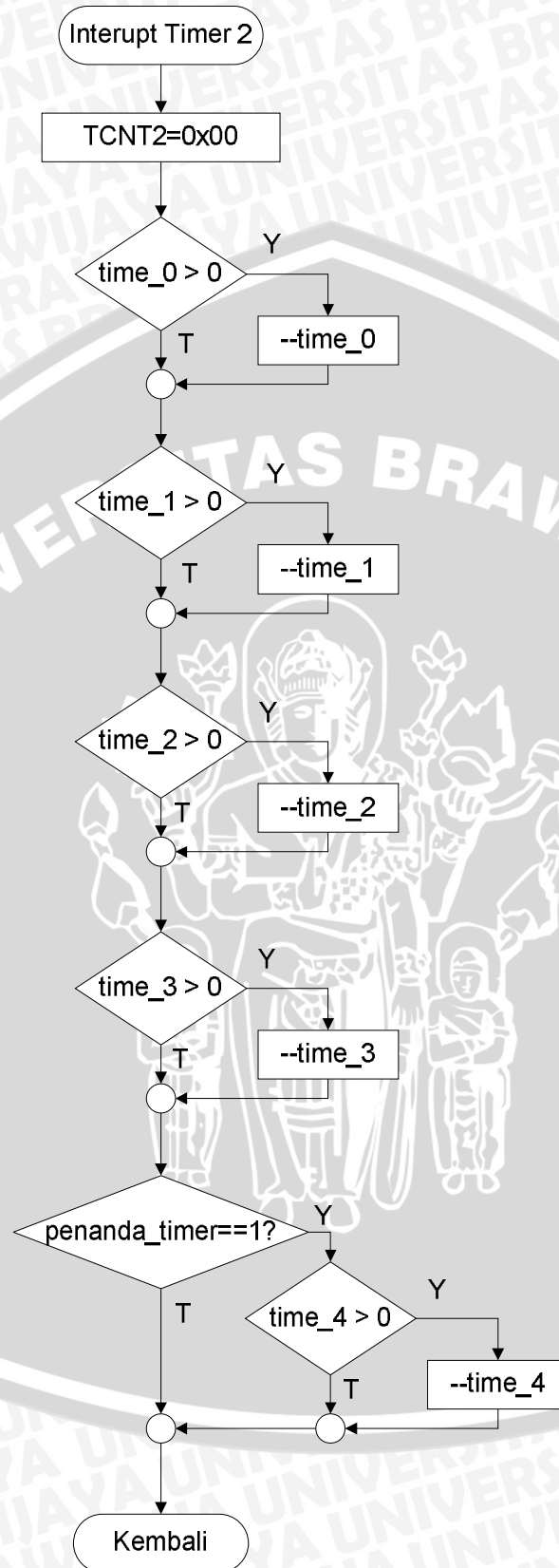
Gambar 4.22 Diagram alir sub rutin pengecekan *limit switch*

4.7.10 Perancangan Sub Rutin *Interrupt* Timer 2

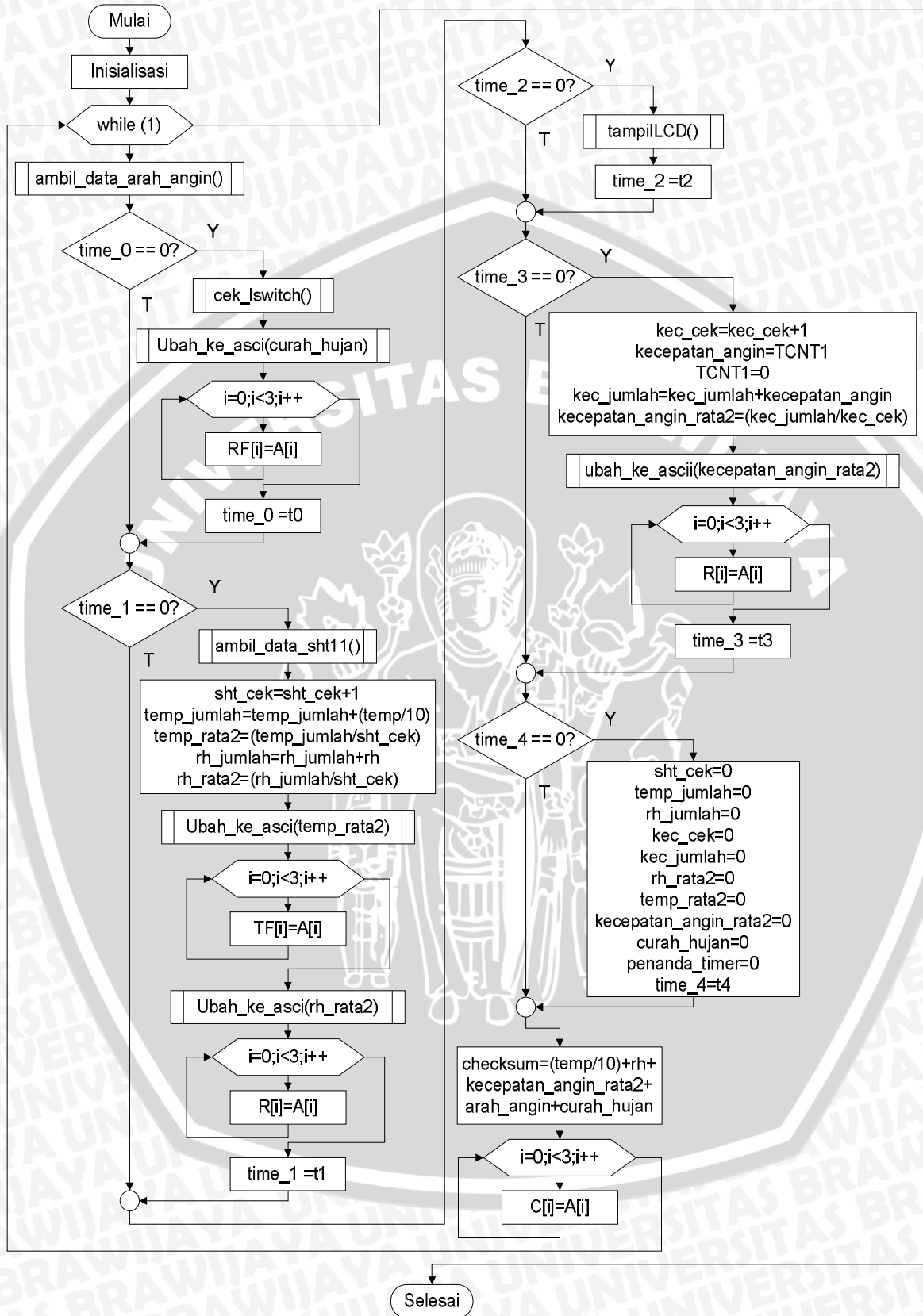
Sub rutin *interrupt* timer 2 berfungsi sebagai pencacah waktu dari proses utama yang meliputi pengambilan data, pengolahan data dan penghapusan data. Pada sub rutin ini nilai dari variabel yang berfungsi sebagai penanda proses utama akan dikurangi tiap satuan waktu timer 2. Apabila penanda waktu sudah bernilai 0, maka proses pada program utama akan dijalankan kemudian pada program utama nilai variabel penanda akan diisi ulang. Diagram alir sub rutin *interrupt* Timer 2 ditunjukkan dalam Gambar 4.23.

4.7.11 Perancangan Rutin Utama

Rutin utama berfungsi untuk menjalankan proses utama yang meliputi pengambilan data, pengolahan data dan penghapusan data. Dalam sub rutin utama, proses ini akan diulang secara terus menerus selama sistem dinyalakan. Diagram alir rutin utam ditunjukkan dalam Gambar 4.24.



Gambar 4.23 Diagram alir sub rutin *interrupt* Timer 2



Gambar 4.24 Diagram alir rutin utama

BAB V

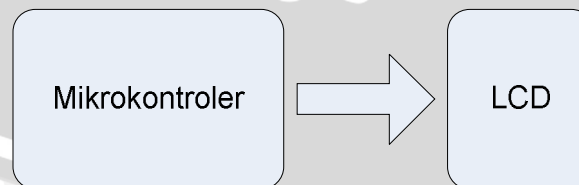
PENGUJIAN DAN ANALISIS

Bab ini akan membahas pengujian dan analisis alat yang telah dirancang dan direalisasikan. Pengujian dilakukan pada tiap-tiap blok dengan tujuan untuk mengamati apakah tiap blok rangkaian sudah sesuai dengan perancangan, baru kemudian dilanjutkan dengan pengujian keseluruhan sistem. Pengujian yang dilakukan sebagai berikut:

- 1) Pengujian modul LCD 16x2
- 2) Pengujian Sensor
 - a) Pengujian rangkaian *limit switch*
 - b) Pengujian rangkaian *Opto*
 - c) Pengujian modul DT-Sense SHT11
 - d) Pengujian pemodelan sensor arah angin
- 3) Pengujian pembangkitan dan transmisi DTMF oleh *handphone*
- 4) Pengujian DTMF *decoder*
- 5) Pengujian pengiriman data secara serial
- 6) Pengujian antarmuka *handphone* dengan mikrokontroler
- 7) Pengujian sistem secara keseluruhan

5.1 Pengujian Modul LCD 16x2

Pengujian modul LCD 16x2 dilakukan dengan menghubungkan LCD dengan mikrokontroler yang sudah berisi perangkat lunak untuk menampilkan tulisan tertentu. Pengujian modul LCD ini bertujuan untuk mengetahui keberhasilan LCD menampilkan tulisan sesuai dengan perangkat lunak yang terdapat dalam mikrokontroler. Diagram blok pengujian modul LCD ini ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Diagram blok pengujian modul LCD

Dalam pengujian ini, LCD dapat menampilkan tulisan "Sistem Pemantau" pada baris pertama dan "Cuaca Jarak Jauh" pada baris kedua. Tampilan hasil pengujian

modul LCD ditunjukkan dalam Gambar 5.2. Dari hasil pengujian dapat dianalisis bahwa rangkaian LCD dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.



Gambar 5.2 Tampilan hasil pengujian modul LCD

5.2 Pengujian Sensor

Pegujian sensor ini bertujuan untuk mengetahui dan menganalisis keberhasilan sensor atau pemodelan sensor mengakuisisi data-data parameter cuaca dan pembacaan data-data dari sensor oleh mikrokontroler. Dalam pengujian ini ada tiga bagian yang akan diuji, yaitu penghitungan penekanan *limit switch*, rangkaian *opto* dan modul DT-Sense SHT11.

5.2.1 Pengujian Penghitungan Penekanan *limit switch*

Pegujian penghitungan penekanan *limit switch* ini bertujuan untuk mengetahui dan menganalisis apakah pemodelan sensor curah hujan yang dimodelkan dengan *limit switch* dapat bekerja sesuai dengan perancangan. Dalam pengujian ini mikrokontroler akan menghitung jumlah penekanan yang terjadi pada limit switch dan menampilkan jumlah penekanan yang terjadi pada modul LCD 16x2. Data yang tampil di LCD tersebut kemudian dibandingkan dengan jumlah penekanan yang dilakukan sebenarnya untuk analisis pengujian. Pengujian ini dilakukan beberapa kali dengan jumlah penekanan yang berbeda-beda. Diagram blok pengujian penghitungan penekanan *limit switch* ditunjukkan dalam Gambar 5.3. Hasil pengujian penghitungan penekanan *limit switch* ditunjukkan dalam Gambar 5.4, Gambar 5.5 dan Tabel 5.1.



Gambar 5.3 Diagram blok pengujian penghitungan penekanan *limit switch*



Gambar 5.4 Hasil pengujian penghitungan penekanan *limit switch* sebanyak 20 kali



Gambar 5.5 Hasil pengujian penghitungan penekanan *limit switch* sebanyak 300 kali

Tabel 5.1 Hasil pengujian penekanan *limit switch*

Jumlah penekanan sebenarnya	Jumlah penekanan yang dihitung mikrokontroler	Persentase kesalahan (%)
20	20	0
30	30	0
50	50	0
100	100	0
200	200	0
300	300	0
Rata-rata kesalahan (%)		0

Dalam pengujian ini, mikrokontroler dapat menghitung jumlah penekanan *limit switch* sesuai dengan jumlah penekanan yang sebenarnya dengan rata-rata kesalahan 0%. Dari hasil pengujian dapat dianalisis bahwa rangkaian *limit switch* dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.

5.2.2 Pengujian Rangkaian *Opto*

Pegujian rangkaian *opto* dibedakan menjadi dua bagian yakni pengujian kecepatan respons rangkaian *opto* dan pengujian jumlah pulsa.

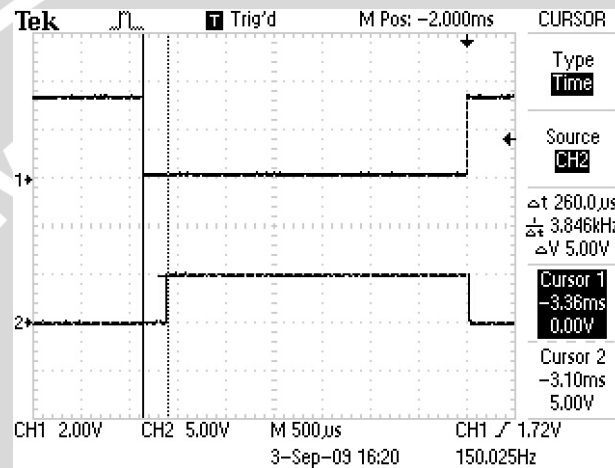
5.2.2.1 Pengujian kecepatan respons

Pengujian ini dilakukan untuk mengetahui dan menganalisis kecepatan maksimal respons rangkaian *opto* yang dirancang sehingga dapat diketahui kecepatan angin maksimal yang dapat diukur menggunakan rangkaian *opto* tersebut. Pengujian dilakukan dengan memberikan pulsa pada LED dengan frekuensi tertentu dan mengamati sinyal keluaran dari *trigger smith* (IC HEF40106BP) dengan menggunakan

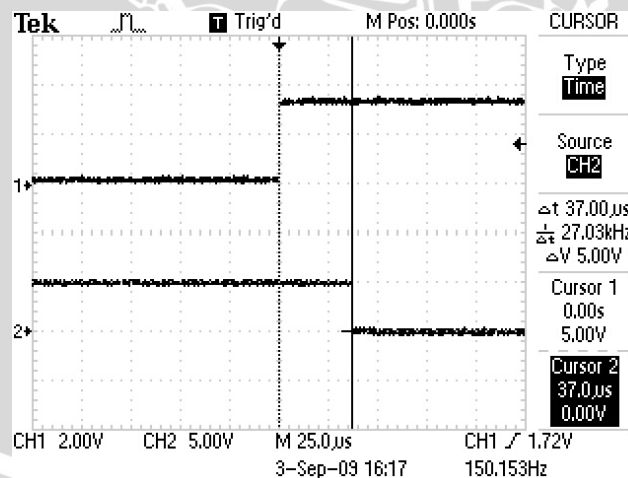
osiloskop. Diagram blok pengujian kecepatan respons sensor opto ini ditunjukkan dalam Gambar 5.6. Hasil pengujian kecepatan respons sensor opto ini ditunjukkan dalam Gambar 5.7 dan Gambar 5.8.



Gambar 5.6 Diagram blok pengujian kecepatan respons sensor opto dalam



Gambar 5.7 Respons keluaran rangkaian opto terhadap masukan pada tepi naik



Gambar 5.8 Respons keluaran rangkaian opto terhadap masukan pada tepi turun

Gambar 5.7 dan Gambar 5.8 menunjukkan hasil pengujian kecepatan respons rangkaian opto terhadap masukan yang diberikan. Ditunjukkan dalam Gambar 5.6, tepi naik pada keluaran terdapat waktu tunda sebesar 260 μ s. Dan dalam Gambar 5.7

ditunjukkan bahwa waktu tunda untuk tepi tuun pada keluaran adalah $37 \mu s$. Dari data hasil pengujian tersebut dapat dianalisis bahwa waktu tunda total (T) adalah:

$$\begin{aligned} T &= (260 + 37) \mu s \\ &= 293 \mu s \end{aligned}$$

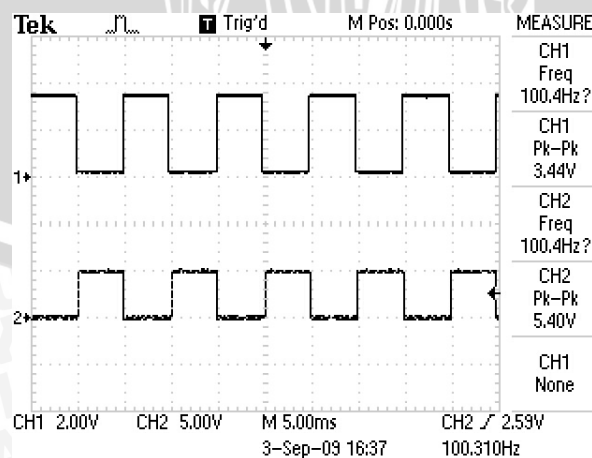
Dengan demikian frekuensi maksimal yang dapat direspons oleh rangkaian opto adalah:

$$\begin{aligned} f_{\text{maks}} &= 1/T \\ &= 1/293 \mu s \\ &= 3,413 \text{ kHz} \end{aligned}$$

Dalam perancangan spesifikasi alat, pemodelan sensor kecepatan angin didesain dengan resolusi 1 km/jam dan nilai maksimal yang dapat diukur adalah 99 km/jam. Sesuai dengan Persamaan 4-4, dengan waktu sampling untuk kecepatan sesaat adalah satu detik, maka kecepatan angin maksimal yang dapat diukur dapat dikonversi menjadi nilai frekuensi. Berdasarkan Persamaan 4-4 maka didapat nilai $n = 99$. Bila diketahui waktu sampling adalah satu detik, maka frekuensi hasil konversi dapat dihitung sebagai berikut:

$$\begin{aligned} f &= n/t \\ &= 99/1 \\ &= 99 \text{ Hz} \end{aligned}$$

Berdasarkan hasil analisis dapat diketahui bahwa frekuensi maksimal yang dapat direspons oleh rangkaian opto lebih besar daripada frekuensi maksimal yang dirancang. Dengan demikian dapat diketahui juga bahwa pemodelan sensor ini dapat bekerja dengan baik pada range pengukuran kecepatan yang dirancang. Hasil pengujian dengan frekuensi 100,4 Hz ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Respons keluaran rangkaian opto terhadap masukan pada frekuensi 100,4 Hz

5.2.2.2 Pengujian penghitungan jumlah pulsa

Pengujian ini dilakukan untuk menganalisis dan mengetahui apakah rangkaian *opto* mampu menghasilkan jumlah pulsa sesuai dengan perancangan. Pengujian ini juga bertujuan untuk mengetahui kesesuaian perangkat lunak mikrokontroler yang berfungsi sebagai penghitung pulsa pemodelan sensor kecepatan angin. Prosedur pengujian dilakukan dengan menghubungkan keluaran rangkaian *opto* dengan mikrokontroler kemudian menghubungkan mikrokontroler yang telah berisi perangkat lunak tersebut dengan modul LCD sesuai dengan Gambar 5.10.



Gambar 5.10 Diagram blok pengujian jumlah pulsa rangkaian *opto*

Mikrokontroler digunakan sebagai penghitung pulsa sensor rotari. Proses perhitungan dilakukan dengan memanfaatkan fasilitas *counter* yang terdapat dalam mikrokontroler. Jumlah pulsa hasil pembacaan *counter* ditampilkan dalam LCD. Pengujian dilakukan dengan memutar piringan model sensor kecepatan angin yang dilengkapi rangkaian *opto* beberapa kali, dengan jumlah putaran yang berbeda-beda.

Hasil pengujian pemutaran piringan sebanyak lima kali atau penghalang melewati rangkaian *opto* sebanyak 20 kali ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Hasil pengujian pemutaran piringan sebanyak lima kali

Hasil pengujian yang diperoleh setelah melakukan beberapa kali pengambilan data ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil pengujian penghitungan pulsa pada pemodelan sensor kecepatan angin

Pengujian ke-	Jumlah putaran	Jumlah pulsa sebenarnya	Jumlah pulsa yang dapat dihitung mikrokontroler	Kesalahan (%)
1	2,5	10	10	0
2	5	20	20	0
3	7,5	30	30	0
4	10	40	40	0
5	12,5	50	50	0
6	15	60	60	0
7	17,5	70	70	0
8	20	80	80	0
9	22,5	90	90	0
10	25	100	100	0
Rata-rata kesalahan (%)				0

Berdasarkan Tabel 5.2, dapat diperoleh hasil bahwa kesalahan rata-rata yang dihasilkan pada pengujian ini adalah 0 %. Dari hasil pengujian dapat dianalisis bahwa rangkaian *opto* dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.

5.2.3 Pengujian modul DT-Sense SHT11

Pengujian ini dilakukan untuk menganalisis dan mengetahui apakah perangkat lunak mikrokontroler mampu mengambil data dari modul DT-Sense SHT11. Prosedur pengujian dilakukan dengan menghubungkan modul DT-Sense SHT11 dengan mikrokontroler sesuai dengan perancangan kemudian menghubungkan mikrokontroler yang telah berisi perangkat lunak tersebut dengan modul LCD sesuai dengan Gambar 5.12.



Gambar 5.12 Diagram blok pengujian modul DT-Sense SHT11

Dalam pengujian ini mikrokontroler berfungsi untuk mengambil data dari Modul DT-Sense SHT11 dan menampilkan data tersebut ke modul LCD. Dalam pengujian ini data yang diambil oleh mikrokontroler adalah data suhu dan kelembaban. Data suhu yang diperoleh mikrokontroler akan diolah terlebih dahulu sehingga diperoleh data suhu dalam satuan °C. Dalam pengujian ini data suhu yang diperoleh dari SHT11 akan dibandingkan dengan nilai suhu yang ditunjukkan termometer yang digunakan sebagai referensi. Sedangkan dalam pengujian pengambilan data kelembaban, data yang

diperoleh akan dibandingkan dengan alat ukur kelembaban merek Dekko (*Dekko 642*). Hasil pengujian pengambilan data suhu dan kelembaban dari Modul DT-Sense SHT11 ditunjukkan dalam Gambar 5.13 dan Gambar 5.14.



Gambar 5.13 Hasil pengujian pengambilan data suhu dari Modul DT-Sense SHT11

Hasil pengujian yang diperoleh setelah melakukan beberapa kali pengambilan data dengan suhu yang berbeda-beda ditunjukkan dalam Tabel 5.3.

Tabel 5.3 Hasil pengujian pengambilan data suhu dari Modul DT-Sense SHT11

Pengujian ke-	Suhu yang terukur di termometer	Suhu hasil pengujian	Kesalahan (%)
1	26	26	0
2	27	27	0
3	28	28	0
4	29	29	0
5	30	30	0
6	31	32	3,23
7	32	33	3,13
Rata-rata kesalahan (%)			0,91



Gambar 5.14 Hasil pengujian pengambilan data kelembaban dari Modul DT-Sense SHT11

Hasil pengujian yang diperoleh setelah melakukan beberapa kali pengambilan data dengan kelembaban yang berbeda-beda ditunjukkan dalam Tabel 5.4.

Tabel 5.4 Hasil pengujian pengambilan data kelembaban dari Modul DT-Sense SHT11

Pengujian ke-	Kelembaban yang terukur di alat ukur kelembaban	Kelembaban hasil pengujian	Kesalahan (%)
1	75	75	0
2	76	76	0
3	77	77	0
4	78	78	0
5	79	79	0
6	80	80	0
7	81	81	0
8	82	81	1,22
9	83	82	1,20
Rata-rata kesalahan (%)			0,27

Berdasarkan Tabel 5.3, dapat diperoleh hasil bahwa kesalahan rata-rata yang dihasilkan pada pengujian pengambilan data suhu adalah 0,91 %. Dan berdasarkan Tabel 5.4, dapat diperoleh hasil bahwa kesalahan rata-rata yang dihasilkan pada pengujian pengambilan data kelembaban adalah 0,27 %. Dari hasil pengujian dapat disimpulkan bahwa Modul DT-Sense SHT11 dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.

5.2.4 Pengujian Pemodelan Sensor Arah angin

Pengujian ini dilakukan untuk menganalisis dan mengetahui apakah perangkat lunak mikrokontroler mampu mengambil data dari pemodelan sensor arah angin. Prosedur pengujian dilakukan dengan menghubungkan pemodelan sensor arah angin yang terdiri dari tiga buah rangkaian *opto* dan sebuah piringan dengan pola tertentu seperti ditunjukkan dalam Gambar 4.3 dengan mikrokontroler sesuai dengan perancangan kemudian menghubungkan mikrokontroler yang telah berisi perangkat lunak tersebut dengan modul LCD . Diagram blok pengujian pemodelan sensor arah angin ini ditunjukkan dalam Gambar 5.15.



Gambar 5.15 Diagram blok pengujian modul DT-Sense SHT11

Dalam pengujian ini mikrokontroler berfungsi untuk mengambil data pemodelan sensor arah angin dan menampilkan data tersebut ke modul LCD. Hasil pengujian pengambilan data pemodelan sensor arah angin ditunjukkan dalam Gambar 5.16.



Gambar 5.16 Hasil pengujian pengambilan data pemodelan sensor arah angin

Dalam Gambar 5.16. ditunjukkan bahwa angka “0” sama dengan arah utara. Angka “0” di sini berarti nilai desimal dari kombinasi “000” (tiga bit data keluaran rangkaian *opto*). Hasil pengujian yang diperoleh setelah melakukan beberapa kali pengambilan data dengan arah yang berbeda-beda ditunjukkan dalam Tabel 5.5.

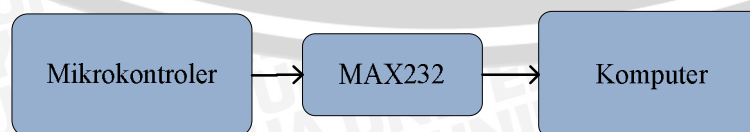
Tabel 5.5 Hasil pengujian pengambilan data pemodelan sensor arah angin

Kombinasi tiga bit data keluaran rangkaian <i>opto</i>	Arah dalam perancangan	Arah hasil pengujian	keterangan
000	Utara (U)	Utara (U)	benar
001	Barat Laut (BL)	Barat Laut (BL)	benar
011	Barat (B)	Barat (B)	benar
010	Barat Daya (BD)	Barat Daya (BD)	benar
110	Selatan (S)	Selatan (S)	benar
111	Tenggara (Tg)	Tenggara (Tg)	benar
101	Timur (T)	Timur (T)	benar
100	Timu Laut (TL)	Timu Laut (TL)	benar
Kesalahan (%)			0

Berdasarkan Tabel 5.5, dapat diperoleh hasil bahwa kesalahan yang dihasilkan pada pengujian ini adalah 0%. Dari hasil pengujian dapat dianalisis bahwa pemodelan sensor arah angin dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.

5.3 Pengujian Pengiriman Data Secara Serial

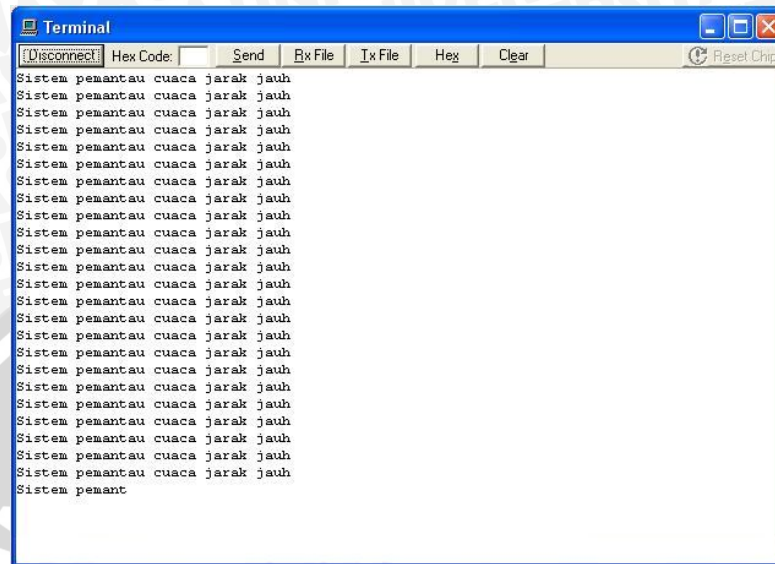
Pengujian pengiriman data secara serial dilakukan dengan menghubungkan mikrokontroler yang sudah berisi perangkat lunak untuk mengirim data secara serial dengan rangkaian MAX232 dan komputer. Pegujian ini bertujuan untuk menganalisis dan mengetahui keberhasilan mikrokontroler mengirim data secara serial. Diagram blok pengujian pengiriman data secara serial ini ditunjukkan dalam Gambar 5.17.



Gambar 5.17 Diagram blok pengujian modul LCD

Dalam pengujian ini komputer berfungsi untuk menerima dan menampilkan data yang dikirimkan mikrokontroler ke monitor melalui terminal yang ada dalam *software*

CodeVision AVR. Dalam pengujian ini mikrokontroler telah berisi perangkat lunak untuk mengirim string "Sistem pemantau cuaca jarak jauh" secara terus-menerus. Hasil pengujian pengiriman data secara serial ditunjukkan dalam Gambar 5.18.



Gambar 5.18 Tampilan hasil pengujian pengiriman data secara serial dalam terminal komputer

Dalam pengujian ini, terminal *software* CodeVision mampu menampilkan tulisan "Sistem pemantau cuaca jarak jauh". Dari hasil pengujian dapat dianalisis bahwa rangkaian mikrokontroler dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.

5.4 Pengujian Antarmuka *Handphone* Dengan Mikrokontroler

Pengujian antarmuka *handphone* dengan mikrokontroler dilakukan dengan menghubungkan *handphone* dengan mikrokontroler yang sudah berisi perangkat lunak untuk mengirim perintah-perintah ke *handphone* dan modul LCD. Pengujian ini bertujuan untuk menganalisis dan mengetahui keberhasilan mikrokontroler mengirim perintah-perintah ke *handphone* dan menerima respons dari *handphone*. Diagram blok pengujian antarmuka *handphone* dengan mikrokontroler ini ditunjukkan dalam Gambar 5.19.



Gambar 5.19 Diagram blok pengujian antarmuka *handphone* dengan mikrokontroler

Dalam pengujian ini mikrokontroler berfungsi untuk mengirimkan dua buah perintah ke *handphone*, yaitu "ate0" yang berfungsi untuk menonaktifkan *echo* pada

handphone dan "at" yang berfungsi untuk mengecek apakah *handphone support* dengan *AT Command* kemudian mengecek apakah respons yang diberikan *handphone* sudah sesuai dengan perancangan. Apabila respons yang diberikan *handphone* benar ("OK"), maka LCD akan menampilkan tulisan "OK" setelah tulisan perintah yang digunakan. Tetapi jika respons dari *handphone* tidak sesuai dengan perancangan (bukan "OK"), maka LCD tidak akan menampilkan tulisan "OK" setelah tulisan perintah yang digunakan. Hasil pengujian antarmuka *handphone* dengan mikrokontroler ditunjukkan dalam Gambar 5.20.



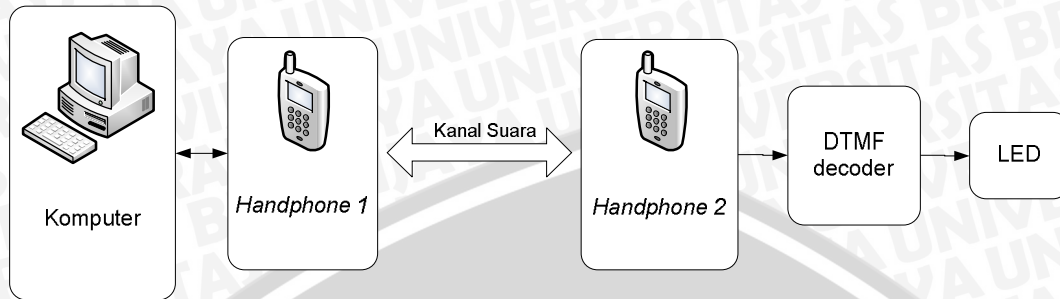
Gambar 5.20 Hasil pengujian antarmuka *handphone* dengan mikrokontroler

Dalam pengujian ini, LCD dapat menampilkan tulisan "OK" setelah tulisan perintah yang dikirim ke *handphone* ("ate0" dan "at"). Dari hasil pengujian dapat dianalisis bahwa rangkaian antarmuka *handphone* dengan mikrokontroler dan perangkat lunak yang telah dibuat dapat bekerja dengan baik.

5.5 Pengujian Penerjemahan Sinyal DTMF yang Dibangkitkan dan Ditransmisikan Melalui Kanal Suara *Handphone*

Pengujian ini dilakukan dengan cara melewatkan sinyal DTMF yang dihasilkan oleh *handphone* pada kanal suara yang terbentuk ketika terjadi panggilan kemudian menerjemahkan sinyal DTMF tersebut menggunakan rangkaian DTMF *decoder* dan melihat nilai biner hasil penerjemahan sinyal DTMF tersebut. Pengujian ini bertujuan untuk menganalisis dan mengetahui apakah sinyal DTMF yang dibangkitkan *handphone* pada RTU dan dikirimkan ke *handphone* yang ada di *Base Station* dapat diterjemahkan sesuai dengan perancangan. Diagram blok pengujian penerjemahan

sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara *handphone* ini ditunjukkan dalam Gambar 5.21.



Gambar 5.21 Diagram blok pengujian penerjemahan sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara *handphone*

Dalam pengujian ini komputer berfungsi untuk memerintahkan *handphone 1* untuk membangkitkan sinyal DTMF sesuai dengan karakter yang diperintahkan komputer. Sinyal DTMF tersebut kemudian ditransmisikan melalui kanal suara dan diterima *handphone 2*, sinyal yang diterima ini kemudian dihubungkan dengan rangkaian DTMF *decoder*. Keluaran rangkaian DTMF *decoder* ini dihubungkan dengan empat buah LED agar nilai biner keluarannya dapat dilihat dan dianalisis. Hasil pengujian penerjemahan sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara *handphone* ini ditunjukkan dalam Tabel 5.6

Tabel 5.6 Hasil pengujian penerjemahan sinyal DTMF yang dibangkitkan dan ditransmisikan melalui kanal suara *handphone*

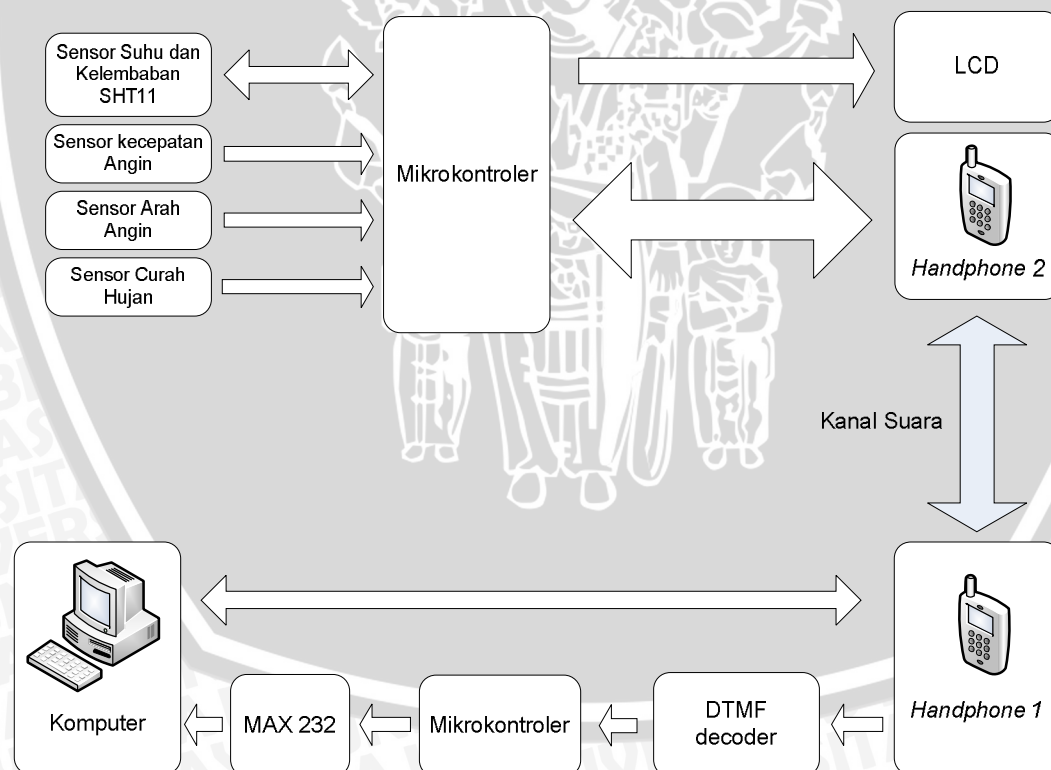
[illegible]

15	AT+VTS=*	*	1	0	1	1	1	0	1	1	benar
16	AT+VTS=#	#	1	1	0	0	1	1	0	0	benar
Kesalahan (%)											0

Berdasarkan Tabel 5.6, dapat diperoleh hasil bahwa kesalahan yang dihasilkan pada pengujian ini adalah 0%. Dari hasil pengujian dapat dianalisis bahwa pembangkitan sinyal DTMF oleh *handphone*, pentransmisian sinyal DTMF melalui kanal suara dan penerjemahan sinyal DTMF menggunakan rangkaian DTMF *decoder* dapat bekerja dengan baik.

5.6 Pengujian Sistem Secara Keseluruhan

Pengujian ini dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat dan memberikan analisis terhadap kinerja alat. Diagram blok pengujian sistem secara keseluruhan ditunjukkan dalam Gambar 5.22.



Gambar 5.22 Diagram blok pengujian sistem secara keseluruhan

Dalam pengujian ini komputer berfungsi untuk memerintahkan *handphone 1* melakukan *dial up* ke *handphone 2*. Ketika menerima panggilan, maka *handphone 2*

memberikan tanda pada mikrokontroler. Mikrokontroler yang telah berisi perangkat lunak sesuai dengan perancangan kemudian akan memerintahkan *handphone* 2 untuk menerima panggilan. Setelah itu data-data parameter cuaca yang tersimpan di mikrokontroler akan dikirimkan melalui kanal suara yang terbentuk di antara *handphone* 1 dan *handphone* 2. Sinyal DTMF yang diterima *handphone* 1 kemudian dihubungkan dengan rangkaian DTMF *decoder*. Keluaran rangkaian DTMF *decoder* ini kemudian dihubungkan dengan mikrokontroler yang berfungsi mengubah data paralel empat bit menjadi karakter-karakter yang sesuai dengan aturan standar DTMF dan mengirimkannya ke komputer secara serial. Untuk menyetarakan level logika antara mikrokontroler dan RS232 digunakan rangkaian MAX232. Data-data yang diterima kemudian akan diolah dan disimpan oleh komputer untuk dianalisis. Hasil pengujian sistem secara keseluruhan ini ditunjukkan dalam Gambar 5.23, Gambar 5.24 dan Gambar 5.25.



Gambar 5.23 Tampilan nilai data yang dikirim dari RTU ke Base Station



Gambar 5.24 Tampilan data yang diterima oleh Base Station

Tabel Parameter Cuaca

Kota Bulan Tahun

Nomer	Waktu	Suhu	Kelembapan	Kecepatan_Angin	Arah_Angin	Curah_Hujan
3	12/3/2009 4:52:48 AM	27	65	5	Utara	25
4	12/3/2009 4:54:48 AM	27	65	5	Utara	25
5	12/3/2009 4:55:48 AM	27	65	5	Utara	25
6	12/3/2009 4:56:48 AM	27	65	5	Utara	25
7	12/3/2009 4:57:48 AM	27	65	5	Utara	25
8	12/3/2009 4:58:48 AM	28	70	10	Utara	30
9	12/3/2009 4:59:48 AM	28	70	10	Utara	30
10	12/3/2009 5:00:48 AM	28	70	10	Utara	30
11	12/3/2009 5:01:48 AM	28	70	10	Utara	30
12	12/3/2009 5:02:48 AM	27	68	4	Tenggara	8

Data Terakhir

C %RH Km/Jam mm

Gambar 5.25 Data dari RTU yang berhasil disimpan dalam basis data

Dari hasil pengujian yang terlihat dalam Gambar 5.22 dan Gambar 5.23, dapat dianalisis bahwa data-data yang dikirimkan RTU (suhu rata-rata, kelembapan rata-rata, curah hujan, kecepatan angin rata-rata dan arah angin) dapat diterima dengan baik oleh *Base Station*. Hal ini terlihat dari kecocokan antara data yang tertulis di LCD dengan data yang tertulis dalam program aplikasi yang ada di *Base Station*. Selain itu dalam Gambar 5.24 juga dapat dilihat bahwa data-data yang diterima oleh *Base Station* telah dapat disimpan ke basis data oleh program aplikasi yang dirancang. Dengan demikian dapat disimpulkan bahwa sistem secara keseluruhan dapat bekerja dengan baik.

Karena sistem ini menggunakan jaringan GSM sebagai media transfer data, maka pengujian ini dilakukan beberapa kali dalam waktu yang berbeda-beda untuk melihat kinerja sistem berdasarkan waktu. Dalam pengujian ini data-data yang diambil adalah waktu, durasi transmisi data, durasi pengambilan data secara keseluruhan, perulangan pengambilan data dan keberhasilan pengiriman data. Data hasil pengujian ini ditunjukkan dalam Tabel 5.7.

Tabel 5.7 Data hasil pengujian sistem secara keseluruhan pada jam-jam tertentu

Pengujian ke-	Jam	Durasi transmisi data	Durasi total pengambilan data	Perulangan pengambilan data	Keberhasilan transmisi data
1	02:57:34	14 detik	24 detik	0	berhasil
2	03:29:38	14 detik	25 detik	0	berhasil
3	04:50:48	14 detik	24 detik	0	berhasil
4	05:02:48	14 detik	24 detik	0	berhasil
5	06:39:52	14 detik	26 detik	0	berhasil
6	07:23:44	14 detik	25 detik	0	berhasil
7	08:53:29	14 detik	58 detik	1	berhasil

8	09:49:16	14 detik	96 detik	2	berhasil
9	10:18:53	14 detik	24 detik	0	berhasil
10	11:39:16	14 detik	61 detik	1	berhasil
11	12:28:16	14 detik	59 detik	1	berhasil
12	13:36:33	14 detik	24 detik	0	berhasil
13	14:42:43	14 detik	24 detik	0	berhasil
14	15:11:27	14 detik	24 detik	0	berhasil
15	16:15:15	14 detik	143 detik	3	berhasil
16	17:15:07	14 detik	60 detik	1	berhasil
18	21:55:49	14 detik	25 detik	0	berhasil
19	22:51:56	14 detik	24 detik	0	berhasil
20	23:12:30	14 detik	24 detik	0	berhasil

Berdasarkan Tabel 5.7, dapat dianalisis bahwa transmisi data antara RTU dan *Base Station* selalu berhasil dalam 20 kali pengujian. Tetapi dalam beberapa pengujian (pengujian ke-7, ke-8, ke-10, ke-11, ke-15, ke-16) terjadi perulangan pengambilan data. Hal ini disebabkan data yang diterima tidak valid, sehingga *Base Station* melakukan *dial up* lagi ke RTU untuk mengambil data. Selain itu dalam Tabel 5.6 ditunjukkan bahwa durasi transmisi data atau pengiriman data dari RTU ke *Base Station* mulai dari terbentuknya kanal suara sampai ditutupnya kanal suara adalah selalu tetap 14 detik. Tetapi pada durasi total pengambilan data atau lama waktu mulai dari komputer memerintahkan *handphone* melakukan *dial up* sampai data valid diterima, kondisinya berbeda-beda. Hal ini disebabkan lamanya waktu panggilan dari *handphone Base Station* untuk masuk ke *handphone* RTU berbeda-beda tergantung kondisi jaringan GSM yang digunakan. Durasi total pengambilan data ini akan semakin lama apabila terjadi perulangan pengambilan data, karena *Base Station* melakukan *dial up* lagi ke RTU untuk mengambil data sehingga dibutuhkan waktu lagi untuk proses ini.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian tiap bagian dan keseluruhan sistem yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- 1) Perancangan sistem elektronika untuk mengakuisisi data-data kecepatan angin, arah angin dan curah hujan dilakukan dengan memodelkan sensor-sensor yang digunakan. Pemodelan yang dilakukan telah berhasil dengan baik menggunakan prinsip kerja sensor-sensor yang sebenarnya secara elektronik dan memberikan data-data yang valid pada saat simulasi. Sedangkan untuk mengakuisisi data-data suhu dan kelembaban digunakan modul DT-Sense SHT11. Modul ini dapat diakses secara serial menggunakan dua jalur koneksi. Data-data yang dihasilkan Modul DT-Sense SHT11 cukup valid dan sesuai dengan kondisi yang sebenarnya.
- 2) Rangkaian antarmuka mikrokontroler dengan sensor-sensor dan *handphone* telah berfungsi dengan baik sehingga mikrokontroler dapat mengambil data-data dari sensor-sensor untuk kemudian diolah dan mengakses *handphone* menggunakan komunikasi serial asinkron. Antarmuka mikrokontroler dengan *handphone* menggunakan pin TX, RX dan GND yang ada pada mikrokontroler dan *handphone* dengan menggunakan *baudrate* 19200 bps.
- 3) Perangkat lunak mikrokontroler yang dirancang berfungsi untuk mengolah data yang diperoleh dari sensor-sensor yang dipakai dan mengakses *handphone* menggunakan *AT Command*. Data-data dari sensor-sensor yang digunakan diambil dan diolah mikrokontroler sesuai dengan pewaktuan yang telah ditentukan. Data-data hasil pengolahan mikrokontroler kemudian dikirimkan melalui kanal suara *handphone* apabila terjadi panggilan pada *handphone*. Data-data yang dikirimkan merupakan *tone-tone* DTMF *handphone* yang dibangkitkan menggunakan perangkat lunak mikrokontroler.
- 4) Resolusi sistem yang dibuat telah teruji dengan benar, yaitu mengacu pada data-data di BMG, 1 °C untuk suhu, 1 %RH untuk kelembaban, 1 km/jam untuk kecepatan angin, 45° untuk arah angin dan 1 mm untuk curah hujan.

- 5) Durasi waktu pengambilan data valid tercepat oleh *Base Station* untuk sekali pengambilan data adalah 24 detik, yang membutuhkan waktu bicara (penggunaan pulsa telpon) selama 14 detik.

6.2 Saran

Saran-saran dalam pengimplementasian maupun peningkatan unjuk kerja sistem ini dapat diuraikan sebagai berikut:

- 1) Saat ini sistem dicatu oleh adaptor yang terhubung dengan sumber listrik jala-jala PLN. Oleh karena itu, disarankan adanya pengembangan catu daya yang lain sehingga sistem tidak bergantung dengan sumber listrik jala-jala PLN, misalkan menggunakan menggunakan *solar cell*.
- 2) Sistem dapat dikembangkan dengan menambahkan media penyimpan data pada RTU, sehingga apabila terjadi kerusakan sistem pada *Base Station*, data-data yang ada di RTU masih bisa diselamatkan.
- 3) Saat ini sensor arah angin, sensor kecepatan angin dan sensor curah hujan yang digunakan dalam sistem masih berupa pemodelan. Oleh karena itu, disarankan adanya pengembangan sistem menggunakan sensor-sensor yang sebenarnya sehingga dapat diterapkan secara langsung di lapangan.

Daftar Pustaka

- Allaby, Michael. 2007. *Encyclopedia Of Weather And Climate, Revised Edition*. New York: Facts On File, Inc.
- Atmel. 2007. *ATMEGA162/ATMEGA162V, 8-bit AVR with 8 Kbytes in System Programmable. Flash*. http://www.atmel.com/dyn/resources/prod_documents/doc2513.pdf. Diakses pada tanggal 15 Maret 2009
- Barry, R.G. dan Chorley, R.J. 2003. *Atmosphere, Weather, and Climate*. London: Roudledge.
- Charrett, Sheldon. 1999. *Electronic Circuits and secrets an Old Fashioned Spy*. Colorado: Paladin Press.
- MITEL. 1995. *ISO²-CMOS MT8870D/MT8870D-1 Integrated DTMF Receiver*. <http://pdf1.alldatasheet.com/datasheet-pdf/view/77085/MITEL/MT8870.html>. Diakses pada tanggal 12 Juli 2009
- Pinouts. 1995. *Siemens A35, A36, A40, C25, C35, C45, M35, M35i, M50, ME45, MT50, S25, S35, S45, SL-42, SL45, 3118 cell phones cable connector pinout*. http://pinouts.ru/CellularPhones-P-W/siemens_c25_s25_pinout.shtml. Diakses pada tanggal 18 November 2009
- Sensirion. 2008. *Datasheet SHT1x (SHT10, SHT11, SHT15) Humidity and Temperature Sensor*. http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT1x.pdf. Diakses pada tanggal 16 Maret 2009
- Siemens. 2000. *AT Command Set Reference Manual*. http://jazi.staff.ugm.ac.id/Mobile%20and%20Wireless%20Documents/s35i_c35i_m35i_atc_commandset_v01.pdf. Diakses pada tanggal 15 Maret 2009
- Smith, Jacqueline. 2006. *The Facts On File Dictionary of Weather and Climate Revised Edition*. New York: Facts On File, Inc.
- Waryono dkk. 1987. *Pengantar Meteorologi Dan Klimatologi Untuk Universitas Dan Umum*. Surabaya: Bina Ilmu.

LAMPIRAN



LAMPIRAN I

FOTO ALAT





Gambar 1 Alat secara keseluruhan



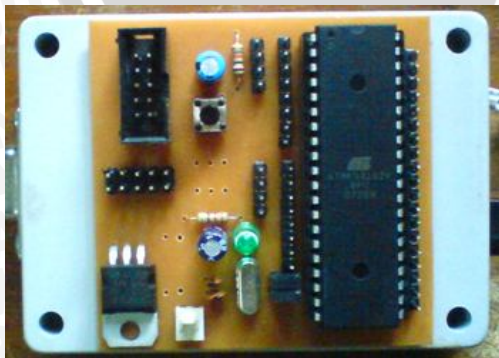
Gambar 2 Pemodelan sensor kecepatan angin dan arah angin



Gambar 3 Kotak yang berisi rangkaian elektronik



Gambar 4 Pemodelan sensor curah hujan
(Tipping Bucket)

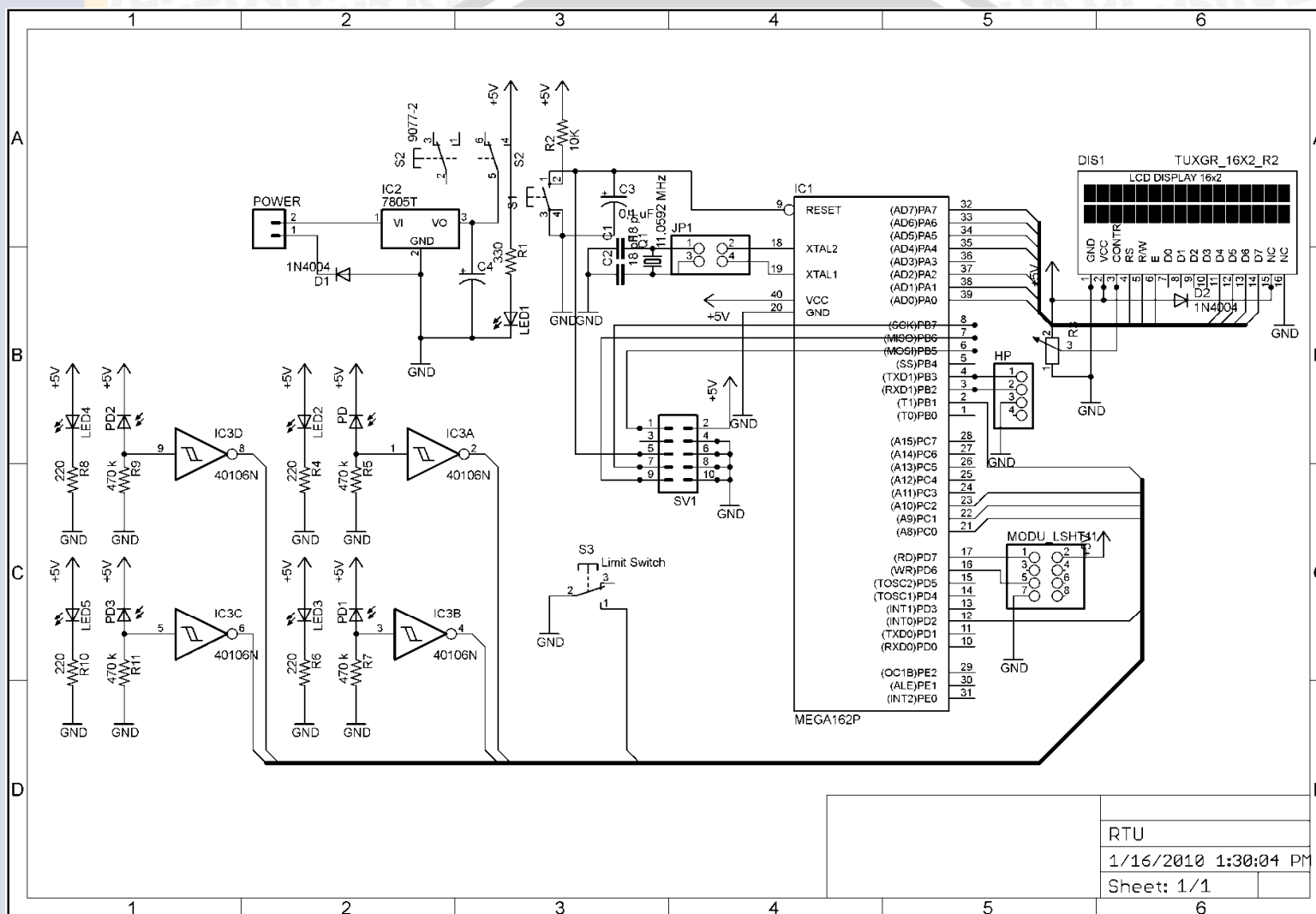


Gambar 5 Rangkaian di dalam sistem

LAMPIRAN II

GAMBAR RANGKAIAN





LAMPIRAN III

LISTING PROGRAM MIKROKONTROLER ATmega162



PROGRAM UTAMA

```
#include <mega162.h>
#include <delay.h>
#include <stdio.h>
#include <string.h>

#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>
#include "sht11/sensirion_protocol.h"
```

```
#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
```

```
#define t0      10
#define t1      2000
#define t2      300
#define t3      1000
#define t4      20000
#define t5      5000
```

```
unsigned long int sht_cek, rh_jumlah, temp_jumlah, kec_cek;
bit penanda_timer=0;
unsigned int checksum, nilai_rh_raw,
nilai_temp_raw,time_0=0,time_1=0,time_2=0,time_3=0, time_4=0, time_5=0,
keadaan_lswitch=1;
```

```
unsigned char error, A[3], B[33],T[3], R[3], MKec[3], Arah[3], RF[3], C[3], rh, rh_rata2,
rh_pengukuran, buff[33];
int temp, temp_rata2, arah_angin, kec_jumlah, kecepatan_angin, kecepatan_angin_rata2,
curah_hujan, i;
```

```
void ubah_ke_ascii (int data)
{
    int data_ratusan, data_puluhan, data_satuan;
```

```
    data_ratusan= data/100;
    A[0]= data_ratusan+48;
    data_puluhan=(data%100)/10;
    A[1]= data_puluhan+48;
    data_satuan=((data%100)%10);
    A[2]=data_satuan+48;
}
```

```
void kirim_data()
{
    printf("AT+VTS=A,2;");
    putchar(13);
    delay_ms (1000);
    printf("AT+VTS=%c,2;",T[1]);
    putchar(13);
    delay_ms (1000);
    printf("AT+VTS=%c,2;",T[2]);
    putchar(13);
    delay_ms (1000);
    printf("AT+VTS=%c,2;",R[1]);
    putchar(13);
    delay_ms (1000);
    printf("AT+VTS=%c,2;",R[2]);
    putchar(13);
    delay_ms (1000);
    // printf("AT+VTS=%c,2;",MKec[0]);
    // putchar(13);
    // delay_ms (1000);
    printf("AT+VTS=%c,2;",MKec[1]);
    putchar(13);
    delay_ms (1000);
    printf("AT+VTS=%c,2;",MKec[2]);
    putchar(13);
}
```

```

delay_ms (1000);
printf("AT+VTS=%c,2;",Arah[2]);
putchar(13);
delay_ms (1000);
// printf("AT+VTS=%c,2;",RF[0]);
// putchar(13);
// delay_ms (1000);
printf("AT+VTS=%c,2;",RF[1]);
putchar(13);
delay_ms (1000);
printf("AT+VTS=%c,2;",RF[2]);
putchar(13);
delay_ms (1000);
printf("AT+VTS=%c,2;",C[0]);
putchar(13);
delay_ms (1000);
printf("AT+VTS=%c,2;",C[1]);
putchar(13);
delay_ms (1000);
printf("AT+VTS=%c,2;",C[2]);
putchar(13);
delay_ms (1000);
printf("AT+VTS=*,2;");
putchar(13);
delay_ms (1000);
}

```

```

void terima_panggilan()
{
delay_ms(3000);
printf ("ATA");
putchar (13);
}

```

```
//=====USART0=====//
```

```

//USART0 Receiver buffer
#define RX_BUFFER_SIZE0 8
char rx_buffer0[RX_BUFFER_SIZE0];

```

```

#if RX_BUFFER_SIZE0<256
unsigned char rx_wr_index0,rx_rd_index0,rx_counter0;
#else

```

```

unsigned int rx_wr_index0,rx_rd_index0,rx_counter0;
#endif

```

```

// This flag is set on USART0 Receiver buffer overflow
bit rx_buffer_overflow0;

```

```

// USART0 Receiver interrupt service routine
interrupt [USART0_RXC] void usart0_rx_isr(void)
{
char status,data;
status=UCSR0A;
data=UDR0;

```

```

if (data=='R')
{
#asm("cli")
penanda_timer=0;
time_4=t4;
terima_panggilan();
delay_ms(1000);
lcd_clear();
lcd_putsf("Kirim Data");
 kirim_data();
 lcd_clear();
 lcd_putsf("Selesai");
 delay_ms(1000);
 penanda_timer=1;
 #asm("sei")
}

```

```

if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer0[rx_wr_index0]=data;
if (++rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
if (++rx_counter0 == RX_BUFFER_SIZE0)
{
rx_counter0=0;
rx_buffer_overflow0=1;
};
};
}

```

```

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART0 Receiver buffer

```



```

#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    char data;
    while (rx_counter0==0);
    data=rx_buffer0[rx_rd_index0];
    if (++rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
    #asm("cli")
    --rx_counter0;
    #asm("sei")
    return data;
}
#pragma used-
#endif

```

```

void cek_lswitch()
{
    switch(keadaan_lswitch)
    {

```

```

        case 1:
            if(PIND.2 == 0)
                keadaan_lswitch=2;
            else
                keadaan_lswitch=1;
            break;

```

```

        case 2:
            if(PIND.2 == 0)
            {
                keadaan_lswitch=3;
                curah_hujan++;
            }
            else
                keadaan_lswitch=1;
            break;

```

```

        case 3:
            if(PIND.2 == 0)
                keadaan_lswitch=3;
            else
                keadaan_lswitch=4;
            break;

```

```

        case 4:
            if(PIND.2 == 0)
                keadaan_lswitch=3;

```

```

            else
            {
                keadaan_lswitch=1;
            }
            break;
        }
    }
}

```

```

void ambil_data_sht11(void)
{

```

```

    error= s_measure(&nilai_temp_raw, 0);
    if(error == 0)
    {
        error= s_measure(&nilai_rh_raw, 1);
    }
    if(error == 0)
    {

```

```

        temp= calc_sht11_temp(nilai_temp_raw);// Hitung suhu (dalam celcius * 10)
        rh= rhcalc_int(nilai_rh_raw);// Hitung kelembapan relatif (dalam 0-100 %)
    }
}

```

```

void tampilLCD(void)
{

```

```

    lcd_clear();
    sprintf(buff, " - %i %i - %i", kecepatan_angin, kecepatan_angin_rata2,
    curah_hujan );
    strcat(B,buff);
    lcd_puts(B);
    sprintf(buff, "\n%i %i%% - %i %i%%C", rh, rh_rata2, temp/10, temp_rata2, 223);
    lcd_puts(buff);
}

```

```

void connect_HP(void)
{

```

```

    lcd_clear();
    lcd_putsf(" Connecting HP");
    delay_ms(1000);
    printf("ATE0");
    putchar(13);
}

```

```

while (getchar()!='O');
if (getchar()=='K')
{
    delay_ms(1000);
    lcd_putsf("\nNow loading...");
};

delay_ms(1000);

printf("AT");
putchar(13);

while(getchar() != 'O');
if (getchar()=='K')
{
    delay_ms(1000);
    lcd_putsf("OK");
}
delay_ms(1000);
}

void ambil_data_arah_angin(void)
{
    if ((PINC.0==0) && (PINC.1==0) && (PINC.2==0))
    {
        arah_angin=0;
        sprintf(B,"U");
    }
    else if ((PINC.0==0) && (PINC.1==0) && (PINC.2==1))
    {
        arah_angin=1;
        sprintf(B,"BL");
    }
    else if ((PINC.0==0) && (PINC.1==1) && (PINC.2==1))
    {
        arah_angin=3;
        sprintf(B,"B");
    }
    else if ((PINC.0==0) && (PINC.1==1) && (PINC.2==0))
    {
        arah_angin=2;
        sprintf(B,"BD");
    }
}

```

```

    }
    else if ((PINC.0==1) && (PINC.1==1) && (PINC.2==0))
    {
        arah_angin=6;
        sprintf(B,"S");
    }
    else if ((PINC.0==1) && (PINC.1==1) && (PINC.2==1))
    {
        arah_angin=7;
        sprintf(B,"Tg");
    }
    else if ((PINC.0==1) && (PINC.1==0) && (PINC.2==1))
    {
        arah_angin=5;
        sprintf(B,"T");
    }
    else if ((PINC.0==1) && (PINC.1==0) && (PINC.2==0))
    {
        arah_angin=4;
        sprintf(B,"TL");
    }
}

ubah_ke_ascii(arah_angin);
for(i=0;i<3;i++)
Arah[i]=A[i];
}

// Timer 2 output compare interrupt service routine
interrupt [TIM2_COMP] void timer2_comp_isr(void)
{
    TCNT2=0x00;
    if(time_0 > 0) --time_0;
    if(time_1 > 0) --time_1;
    if(time_2 > 0) --time_2;
    if(time_3 > 0) --time_3;
    if(penanda_timer==1)
    {
        if(time_4 > 0) --time_4;
    }
    if(time_5 > 0) --time_5;
}

```

```

void main(void)
{
// Crystal Oscillator division factor: 1
#pragma optsize-
CLKPR=0x80;
CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif

```

```

PORTA=0x00;
DDRA=0x00;

```

```

PORTB=0x01;
DDRB=0x00;

```

```

PORTC=0x00;
DDRC=0x00;

```

```

PORTD=0x04;
DDRD=0x00;

```

```

PORTE=0x00;
DDRE=0x00;

```

```

TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

```

```

TCCR1A=0x00;
TCCR1B=0x07;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

```

```

ASSR=0x00;
TCCR2=0x04;
TCNT2=0x00;
OCR2=0xAC;

```

```

TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;

```

```

MCUCR=0x00;
EMCUCR=0x00;

```

```

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;
ETIMSK=0x00;

```

```

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 19200

```

```

UCSR0A=0x00;
UCSR0B=0x98;
UCSR0C=0x86;
UBRR0H=0x00;
UBRR0L=0x23;

```

```

ACSR=0x80;

```

```

lcd_init(16);

```

```

// Global enable interrupts
#asm("sei")

```

```

time_0=t0;
time_1=t1;
time_2=t2;
time_3=t3;
time_4=t4;

```



```

lcd_clear();
lcd_putsf("Sistem Pemantau\nCuaca Jarak Jauh");
for (i=0;i<20;i++)
{
    ambil_data_sht11();
}
connect_HP();

while (1)
{
    printf ("A");
    ambil_data_arah_angin();

    if(time_0 == 0) // tiap 10 ms
    {
        cek_lswitch();
        ubah_ke_ascii(curah_hujan);
        for(i=0;i<3;i++)
            RF[i]=A[i];
        time_0=t0;
    }

    if(time_1 == 0) // tiap 2 s
    {
        ambil_data_sht11();
        sht_cek=sht_cek+1;
        temp_jumlah=temp_jumlah+(temp/10);
        temp_rata2=(temp_jumlah/sht_cek);

        rh_jumlah=rh_jumlah+rh;
        rh_rata2=(rh_jumlah/sht_cek);

        ubah_ke_ascii(temp_rata2);
        for(i=0;i<3;i++)
            T[i]=A[i];

        ubah_ke_ascii(rh_rata2);
        for(i=0;i<3;i++)
            R[i]=A[i];

        time_1=t1;
    }

    if(time_2 == 0) // tiap 300 ms

```

```

{
    tampilLCD();
    time_2=t2;
}

if(time_3 == 0) // tiap 1 s
{
    kec_cek=kec_cek+1;
    kecepatan_angin=TCNT1;
    TCNT1=0;
    kec_jumlah=kec_jumlah+kecepatan_angin;
    kecepatan_angin_rata2=(kec_jumlah/kec_cek);
    ubah_ke_ascii(kecepatan_angin_rata2);
    for(i=0;i<3;i++)
        MKec[i]=A[i];
    time_3=t3;
}

if(time_4 == 0)
{
    sht_cek=0;
    temp_jumlah=0;
    rh_jumlah=0;
    kec_cek=0;
    kec_jumlah=0;
    rh_rata2=0;
    temp_rata2=0;
    kecepatan_angin_rata2=0;
    curah_hujan=0;

    penanda_timer=0;
    time_4=t4;
}

checksum=(temp/10)+rh+kecepatan_angin_rata2+arah_angin+curah_hujan;
ubah_ke_ascii(checksum);
for(i=0;i<3;i++)
    C[i]=A[i];
};
}

```

sensirion_protocol.c

```
#include <io.h>
#ifndef F_CPU
#define F_CPU 4000000UL // 4 MHz
#endif
#include <delay.h>
```

```
// Command untuk sht11
#define STATUS_REG_W 0x06 //000 0011 0
#define STATUS_REG_R 0x07 //000 0011 1
#define MEASURE_TEMP 0x03 //000 0001 1
#define MEASURE_HUMI 0x05 //000 0010 1
#define RESET 0x1e //000 1111 0
```

```
#define SETSCK1 PORTD|=(1<<6)
#define SETSCK0 PORTD&=~(1<<6)
#define SCKOUTP DDRD|=(1<<6)
// ---
#define SETDAT1 PORTD|=(1<<7)
#define SETDAT0 PORTD&=~(1<<7)
#define GETDATA (PIND&(1<<7))
// ---
#define DMODEIN DDRD&=~(1<<7)
#define PULLUP1 PORTD|=(1<<7)
#define DMODEOU DDRD|=(1<<7)
```

```
// Delay-delay
#define S_PULSLONG delay_us(3)
#define S_PULSSHORT delay_us(1)
```

```
// Kirim 1 byte data dan cek acknowledge
char s_write_byte(unsigned char value)
```

```
{
    unsigned char i= 0x80;
    unsigned char error= 0;

    DMODEOU;
    while(i)
    {
        //shift bit for masking
        if(i & value)
```

```
    {
        SETDAT1; //masking value with i , write to SENSI-BUS
    }else
    {
        SETDAT0;
    }
    SETSCK1; //clk for SENSI-BUS
    S_PULSLONG;
    SETSCK0;
    S_PULSSHORT;
    i= (i>>1);
}
DMODEIN; //release DATA-line
PULLUP1;
SETSCK1; //clk #9 for ack
S_PULSLONG;
if(GETDATA)
{
    //check ack (DATA will be pulled down by SHT11)
    error= 1;
}
S_PULSSHORT;
SETSCK0;
return(error); //error=1 in case of no acknowledge
}

// reads a byte form the Sensibus and gives an acknowledge in case of "ack= 1"
// reversebits= 1 caused the bits to be reversed (bit0= bit7, bit1= bit6,...)
unsigned char s_read_byte(unsigned char ack)
{
    unsigned char i= 0x80;
    unsigned char val= 0;
    DMODEIN; //release DATA-line
    PULLUP1;
    while(i)
    {
        //shift bit for masking
        SETSCK1; //clk for SENSI-BUS
        S_PULSSHORT;
        if(GETDATA)
        {
            val= (val | i); //read bit
        }
        SETSCK0;
    }
}
```

```

    S_PULSSHORT;
    i= (i>>1);
}
DMODEOU;
if(ack)
{
    //in case of "ack==1" pull down DATA-Line
    SETDAT0;
}
else
{
    SETDAT1;
}
SETSCK1; //clk #9 for ack
S_PULSLONG;
SETSCK0;
S_PULSSHORT;
DMODEIN; //release DATA-line
PULLUP1;
return (val);
}

// This is the point where sensirion is not I2C standard conform and the
// main reason why the AVR TWI hardware support can not be used.
//
// DATA:  _____
//
// SCK :  _____
void s_transstart(void)
{
    // Initial state
    SCKOUTP;
    SETSCK0;
    DMODEOU;
    SETDAT1;

    S_PULSSHORT;
    SETSCK1;
    S_PULSSHORT;
    SETDAT0;
    S_PULSSHORT;
    SETSCK0;
    S_PULSLONG;
    SETSCK1;

```

```

    S_PULSSHORT;
    SETDAT1;
    S_PULSSHORT;
    SETSCK0;
    S_PULSSHORT;
    DMODEIN; //release DATA-line
    PULLUP1;
}

// makes a measurement (humidity/temperature) with checksum
// p_value returns 2 bytes
// mode: 1= humidity 0= temperature
// return value: 1= write error, 2= timeout
//
char s_measure(unsigned int *p_value, unsigned char mode)
{
    unsigned char i= 0;
    unsigned char msb, lsb;

    *p_value= 0;
    s_transstart(); //transmission start
    if(mode)
    {
        mode= MEASURE_HUMI;
    }
    else
    {
        mode= MEASURE_TEMP;
    }
    if(s_write_byte(mode))
    {
        return(1);
    }
    //crc_state= computeCRC8(bitswapbyte(mode),crc_state);

    // normal delays: temp i=70, humi i=20
    while(i < 240)
    {
        delay_ms(3);
        if(GETDATA == 0)
        {
            i=0;
            break;
        }
    }
}

```



```

    }
    i++;
};

if(i)
{
    // or timeout
    return(2);
}

msb= s_read_byte(1); //read the first byte (MSB)
lsb= s_read_byte(1); //read the second byte (LSB)
*p_value= ((unsigned int) msb<<8)|(lsb);
return(0);
}

// calculates temperature [C]
// input : temp [Ticks] (14 bit)
// output: temp [C] times 10 (e.g 253 = 25.3'C)
// Sensor supply voltage: about 5V
//
int calc_sth11_temp(unsigned int t)
{
    //t = 10*(t *0.01 - 40.1);
    //or
    //t = t *0.1 - 401;
    t/= 10;
    t-= 401;
    return(t);
}

// calculates humidity [%RH]
// The relative humidity is: -0.0000028*s*s + 0.0405*s - 4.0
// but this is very difficult to compute with integer math.

```

```

// We use a simpler approach. See sht10_Non-
// Linearity_Compensation_Humidity_Sensors_E.pdf
// output: humi [%RH] (=integer value from 0 to 100)
unsigned char rhcalc_int(unsigned int s)
{
    // s is in the range from 100 to 3340
    unsigned int rh;

    //for s less than 1712: (143*s - 8192)/4096
    //for s greater than 1712: (111*s + 46288)/4096
    //s range: 100<s<3350

    //
    //rh = rel humi * 10
    if(s < 1712)
    {
        // div by 4:
        rh= (36*s - 2048)/102;
    }
    else
    {
        // div by 8:
        rh= (14*s + 5790)/51;
    }

    // round up as we will cut the last digit to full percent
    rh+= 5;
    rh/= 10;

    if(rh > 98)
    {
        rh= 100;
    }

    return((unsigned char)(rh));
}

```