

**RANCANG BANGUN
SISTEM PENGUKURAN TERDISTRIBUSI
BERBASIS JARINGAN MULTIDROP RS-485**

TUGAS AKHIR

**Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Fisika**

Oleh :

M. Dimas Khoirul Fahmi

0210930026



**JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2008**

LEMBAR PENGESAHAN

RANCANG BANGUN
SISTEM PENGUKURAN TERDISTRIBUSI
BERBASIS JARINGAN MULTIDROP RS-485

Oleh :
M. DIMAS KHOIRUL FAHMI
0210930026-93

Setelah dipertahankan di depan Majelis Penguji
pada tanggal.....
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Fisika

Telah diperiksa dan disahkan oleh :

Menyetujui,
Pembimbing I,

Pembimbing II,

Dr. Eng. Didik R. Santoso, M.Si.
NIP. 132 086 158

Ahmad Nadhir, S.Si., M.T.
NIP. 132 232 182

Mengetahui,
Ketua Jurusan Fisika
Fakultas MIPA Universitas Brawijaya

Drs. Adi Susilo, M.Si, Ph.D.
NIP. 131 960 447

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : M. Dimas Lhoirul Fahmi
NIM : 0210930026
Jurusan : Fisika
Penulis Skripsi :

RANCANG BANGUN SISTEM PENGUKURAN TERDISTRIBUSI BERBASIS JARINGAN MULTIDROP RS-485

Dengan ini menyatakan bahwa :

1. Skripsi ini adalah benar-benar karya saya sendiri, dan bukan hasil jiplakan dari karya orang lain. Karya-karya yang tercantum dalam Daftar Pustaka Skripsi ini, semata-mata digunakan sebagai acuan atau referensi.
2. Apabila dikemudian hari diketahui bahwa isi Skripsi saya merupakan hasil jiplakan, maka saya akan bersedia menanggung segala resiko dari keadaan tersebut.
Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, April 2008
Yang menyatakan,

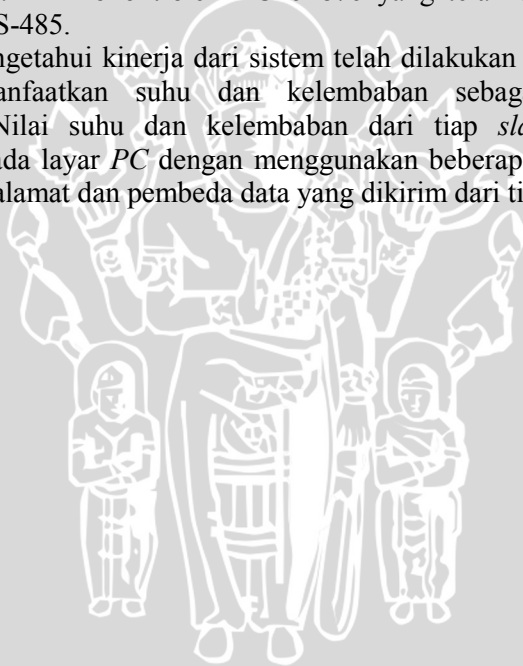
(M. Dimas Khoirul Fahmi)
NIM. 0210930026

RANCANG BANGUN SISTEM PENGUKURAN TERDISTRIBUSI BERBASIS JARINGAN MULTIDROP RS-485

ABSTRAK

Sistem terdistribusi merupakan sistem yang biasa digunakan untuk komunikasi data dengan satu pusat sistem sebagai master dan beberapa percabangan sebagai *slave*. Pada penelitian ini suatu sistem pengukuran terdistribusi berbasis jaringan multidrop RS-485 telah berhasil dibuat. Master sebagai pusat pengendali serta pengolah data berupa *PC* yang telah dilengkapi RS-232 To RS-485 *Converter*. Sedangkan *slave* berfungsi untuk mengirimkan data hasil pengukuran yang terdiri dari mikrokontroler PIC16F876 yang telah dilengkapi dengan NIC RS-485.

Untuk mengetahui kinerja dari sistem telah dilakukan pengujian dengan memanfaatkan suhu dan kelembaban sebagai obyek pengukuran. Nilai suhu dan kelembaban dari tiap *slave* dapat ditampilkan pada layar *PC* dengan menggunakan beberapa karakter kunci sebagai alamat dan pembeda data yang dikirim dari tiap *slave*.



Development of Distributed Measurement System Based on RS-485 Multidrop Network

ABSTRACT

A distributed system is a data communication system comprises a single master and multi slaves. In this research, a distributed measurement system based on RS-485 multidrop network has been built succesfully. The master is a data processor, that is a PC equipped with a RS-232 To RS-485 Converter. The slave, that consists of a microcontroller PIC16F876 with a NIC RS-485, transfers the measurement data.

The performance of the system was tested in the experiments using temperature and humidity as the measurement parameters. The results show that the temperature and humidity from each slave can be displayed on a PC monitor with some key characters as the address from each slave.



KATA PENGANTAR

Berkat rahmat Allah Yang Maha Kuasa, puji dan syukur penulis panjatkan pada-Nya sehingga penulis dapat menyelesaikan penulisan tugas akhir berjudul **“RANCANG BANGUN SISTEM PENGUKURAN TERDISTRIBUSI BERBASIS JARINGAN MULTIDROP RS-485”**. Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana sains dalam bidang Fisika di Fakultas MIPA Universitas Brawijaya Malang.

Penulis banyak mendapat bantuan dan dukungan dari berbagai pihak dalam penyusunan tugas akhir ini, sehingga pada kesempatan ini penulis ucapkan banyak terima kasih kepada:

1. Kedua orang tua, H. Suyuthi Asyrof dan Hj. Masruroh Fahmi yang selalu melecut semangat dan mendoakan. Tanpa restu beliau berdua takkan ada kesuksesan.
2. Dr. Eng Didik R Santoso, M.Si sebagai pembimbing I atas segala bentuk bimbingan yang beliau berikan selama penyusunan tugas akhir.
2. Ahmad Nadhir, S.Si, M.T, sebagai pembimbing II atas segala bentuk bimbingan yang beliau berikan selama penyusunan tugas akhir.
3. Drs. Adi Susilo selaku Ketua Jurusan Fisika.
4. Bapak dan ibu dosen staf pengajar, laboran, dan karyawan jurusan Fisika Universitas Brawijaya.
5. Kakak-kakak dan Adik-adikku yang selalu mendukung dan mendoakan.
6. Teman-teman jurusan FISIKA 2002 (generasi kreatif, inovatif dan sedikit jorok).
7. Semua pihak yang telah mendukung dan membantu baik langsung maupun tidak langsung.

Akhir kata penulis memohon maaf sebesar-besarnya atas segala bentuk kekurangan dalam penyusunan tugas akhir ini.

Malang, April 2008

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
ABSTRAK/ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	viii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	
2.1 Sistem Terdistribusi dan Jaringan Multidrop RS-485	4
2.2 <i>Device Network Interface Card(NIC)</i> RS-485	5
2.3 Mikrokontroler PIC16F876.....	6
2.4 Port Serial Komputer	8
2.5 Sensor Suhu.....	10
2.6 Sensor Kelembaban.....	11
BAB III METODOLOGI	
3.1 Waktu dan Tempat Penelitian	12
3.2 Tahap Pengerjaan	12
3.3 Perancangan Perangkat Keras	12
3.3.1 Perancangan Modul Master.....	13
3.3.2 Perancangan Modul Slave	16
3.3.3 Perancangan Rangkaian Sensor.....	19
a. Rancangan Sensor Suhu.....	19
b. Rangkaian Sensor Kelembaban.....	21

3.3.4 Perancangan Protokol Komunikasi	24
3.4 Pengujian Sistem Peralatan.....	28
3.4.1 Pengujian Master.....	28
3.4.2 Pengujian Modul Slave.....	29
3.4.3 Pengujian Rangkaian Sensor	30
3.4.4 Pengujian Sistem Keseluruhan.....	31

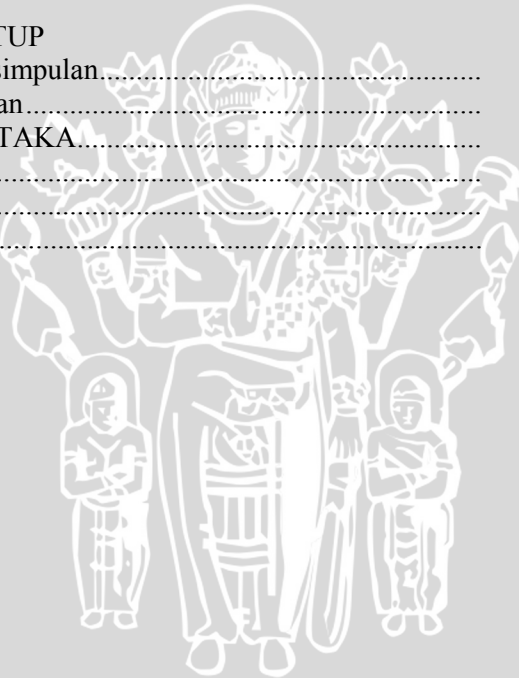
BAB IV HASIL dan PEMBAHASAN

4.1 Hasil Pengujian alat.....	32
a. Pengujian Sensor Suhu.....	32
b. Pengujian Sensor Kelembaban.....	33
4.2 Pengujian Software.....	35

BAB V PENUTUP

5.1 Kesimpulan.....	44
5.2 Saran.....	44

DAFTAR PUSTAKA.....	45
LAMPIRAN 1	46
LAMPIRAN 2	48
LAMPIRAN 3	65



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Sistem Terdistribusi	4
Gambar 2.2 Blok Diagram RS-485.....	6
Gambar 2.3 Mikrokontroler PIC16F876.....	7
Gambar 2.4 Pengiriman data asinkron.....	8
Gambar 2.5 Konfigurasi pin DB9.....	9
Gambar 2.6 DB9 <i>Male</i>	9
Gambar 2.7 DB9 <i>Female</i>	9
Gambar 2.8 Pemasangan DB9 <i>Female</i> dan DB9 <i>Male</i>	10
Gambar 2.9 Sensor Kelembaban HS15P.....	11
Gambar 3.1 Perancangan perangkat keras.....	12
Gambar 3.2 Rangkaian RS-485 To RS-232 <i>Converter</i>	14
Gambar 3.3 Blok diagram RS-232 To RS-485 <i>Converter</i>	15
Gambar 3.4 Rangkaian modul <i>Slave</i>	16
Gambar 3.5 Prosedur inisialisasi.....	17
Gambar 3.6 Prosedur Konversi.....	18
Gambar 3.7 Rangkaian sensor suhu.....	19
Gambar 3.8 Rangkaian Osilator <i>Wien</i>	21
Gambar 3.9 Rangkaian jembatan <i>Wheatstone</i>	22
Gambar 3.10 Rangkaian penguat diferensial.....	23
Gambar 3.11 Rangkaian penyearah AC/DC.....	24
Gambar 3.12 Contoh transfer data pada Slave 1	26
Gambar 3.13 Diagram alir transfer data tiga <i>slave</i>	27
Gambar 4.1 Grafik hasil pengujian sensor suhu.....	33
Gambar 4.2 Keluaran rangkaian osilator <i>Wien</i>	34
Gambar 4.3 Grafik hasil pengujian jembatan <i>Wheatstone</i>	34
Gambar 4.4 Grafik hasil pengujian sensor kelembaban.....	35
Gambar 4.5 Tampilan <i>software</i>	36
Gambar 4.6 Grafik nilai suhu dari <i>slave</i> 1.....	37
Gambar 4.7 Grafik nilai kelembaban dari <i>slave</i> 1.....	38
Gambar 4.8 Grafik nilai suhu dari <i>slave</i> 2.....	38
Gambar 4.9 Grafik nilai kelembaban dari <i>slave</i> 2.....	39
Gambar 4.10 Grafik nilai suhu dari <i>slave</i> 3.....	39
Gambar 4.11 Grafik nilai kelembaban dari <i>slave</i> 3.....	40
Gambar 4.12 Topologi linear dua kabel untuk RS-485.....	42
Gambar 4.13 Multipoint dan individual point.....	43

DAFTAR LAMPIRAN

Lampiran 1

- 1.1 Data Uji Jembatan WheatStone
- 1.2 Data Uji Penguat Sensor Suhu
- 1.3 Data Uji Sensor Suhu
- 1.4 Data Uji Sensor Kelembaban

Lampiran 2

- 2.1 *Source Code Master* (Delphi7)
- 2.2 *Source Code* Mikrokontroler

Lampiran 3

- 3.1 Data pengujian *software* kelembaban
- 3.2 Data pengujian *software* suhu



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengukuran dan monitoring parameter fisis merupakan pekerjaan yang sering dilakukan oleh manusia. Pengukuran merupakan kegiatan yang bertujuan untuk mengetahui kondisi fisis suatu obyek baik dimensi (seperti luas) maupun sifat fisis internal (seperti suhu) yang dimiliki obyek tersebut. Monitoring merupakan suatu kegiatan yang bertujuan untuk mengamati dan mengawasi segala bentuk perubahan kondisi fisis suatu obyek.

Pengukuran dan monitoring terkadang dilakukan pada obyek pengukuran yang tersebar di sejumlah tempat dengan jarak berjauhan dan terkadang membahayakan. Rumah sakit sebagai salah satu contoh obyek pengukuran tersebar memiliki sejumlah ruangan dengan kondisi fisis yang berbeda bergantung pada fungsi dari ruangan tersebut. Kondisi fisis (seperti suhu, kelembaban dan tekanan udara) tersebut akan selalu dipantau selama 24 jam oleh pihak rumah sakit untuk menunjang dan mendukung proses penyembuhan pasien. Pengamatan cuaca sebagai contoh lain dari pengukuran yang tersebar juga memiliki kebutuhan yang sama. Penempatan sarana pengamatan cuaca yang letaknya tersebar akan merepotkan apabila tidak dapat diamati dari satu tempat. Terkadang penempatan sarana pengamatan cuaca tersebut terletak pada daerah yang cukup sulit dijangkau oleh manusia dan membahayakan.

Proses perekaman dan penyimpanan data secara manual memiliki beberapa kelemahan. Media pencatatan yang masih memanfaatkan kertas memiliki kekurangan dalam kapasitas penyimpanan dan masa pakai media penyimpanan tersebut. Perbedaan bentuk tulisan dari pencatatan secara manual juga menjadi masalah tersendiri pada saat dilakukan pengamatan ulang hasil rekaman pengukuran yang telah dicatat.

Melihat fenomena di atas, pada penelitian ini kami mencoba memberikan gagasan dengan membuat perangkat yang dapat memonitor besaran fisis lingkungan seperti suhu dan kelembaban selama 24 jam nonstop meskipun ruangan yang dimonitor berjumlah banyak. Dengan sistem terdistribusi multidrop RS-485, penempatan beberapa *slave* pada beberapa tempat dapat memberikan informasi kondisi fisis suatu tempat. Hasil pengamatan dapat diamati dan

disimpan pada *PC* untuk menjaga tingkat efisiensi. Monitoring dilakukan melalui satu *PC* dengan memperbanyak jumlah *slave* dan sensor yang dibutuhkan pada tempat yang akan dimonitor.

1.2 Perumusan Masalah

Permasalahan yang ada pada penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang sistem terdistribusi berbasis jaringan Multidrop RS-485.
2. Bagaimana merancang dan membuat modul master termasuk data *logger* untuk sistem fisis yang akan diukur.
3. Bagaimana merancang dan membuat modul *slave* untuk sistem akuisisi data dari sistem fisis yang akan diukur.
4. Bagaimana merancang dan membuat rangkaian sensor suhu dan kelembaban udara untuk contoh aplikasi.
5. Bagaimana membuat protokol komunikasi antara *master* dan *slave*.
6. Bagaimana membuat *software* untuk menampilkan hasil pengukuran pada komputer.

1.3 Batasan Masalah

- Pengukuran suhu dan kelembaban hanya sebagai contoh aplikasi dari sistem yang dibuat.
- Pada penelitian ini hanya dibuat tiga buah *slave*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan membuat suatu sistem pengukuran berbasis jaringan multidrop RS-485. Sistem pengukuran yang telah dibuat dapat mengukur besaran suhu dan kelembaban relatif di beberapa tempat secara terintegrasi dan dapat dimonitor dari satu tempat. Data hasil pengukuran suhu dan kelembaban relatif dari tiap-tiap tempat (sebagai fungsi waktu) dapat disimpan pada satu pusat penyimpanan data untuk keperluan lebih lanjut.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

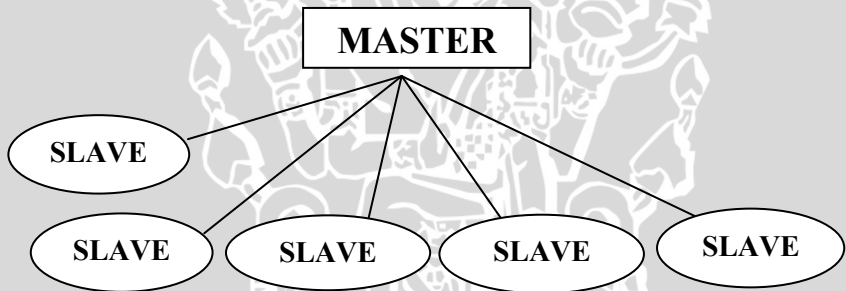
1. Dihasilkan alat monitoring dan data logger suhu dan kelembaban relatif ruangan yang praktis.
2. Dengan adanya alat ini diharapkan dapat memberikan kemudahan dan efektifitas dalam bidang terkait dalam rangka monitoring suhu dan kelembaban.



BAB II TINJAUAN PUSTAKA

2.1 Sistem Terdistribusi dan Jaringan Multidrop RS-485

Sistem terdistribusi adalah sebuah sistem jaringan yang memiliki sifat koneksi terbuka (*open connection*) dan terbagi-bagi atau tersebar. Sifat koneksi terbuka yang dimiliki oleh sistem terdistribusi menunjukkan kemampuan dari sistem tersebut dapat dikomunikasikan dengan perangkat lain. Sistem terdistribusi terdiri dari unit utama (*central unit*) sebagai pengendali sistem secara keseluruhan dan unit bagian (*distributed unit*) sebagai bagian dari sistem yang dikendalikan oleh unit utama meskipun posisi dari unit bagian tersebar dan jauh dari bagian utama. Sistem terdistribusi memungkinkan beberapa sensor dapat dinaungi atau diolah oleh sistem tersebut dalam satu payung atau satu pusat sistem kerja (Lyons, 1999).



Gambar 2.1 Contoh sistem terdistribusi dengan satu *master* sebagai pengendali sistem dan lima *slave* sebagai unit bagian yang dikendalikan oleh *master*.

Jaringan *MultiDrop RS-485* merupakan sistem jaringan yang terdiri dari satu terminal utama sebagai pengontrol dan memiliki percabangan-percabangan yang terpisah sebagai bagian yang berada secara langsung pada tempat *monitoring* dilakukan, namun dalam bekerjanya tetap berada dalam naungan terminal utama. Apa yang diperoleh dari tiap-tiap cabang tersebut kemudian dikirimkan menuju terminal utama untuk diolah lebih lanjut.

Sistem jaringan terdistribusi dapat dilakukan dengan memanfaatkan *network interface* yang dikenal sebagai RS-485.

Interface ini dirancang untuk sebagai *network interface* yang dapat terpecah hingga 32 titik yang terpenjar. Namun agar dapat dikomunikasikan dengan sebuah *PC*, perlu dirubah dulu menjadi komunikasi yang memanfaatkan RS-232 yang umum dipakai oleh *PC* saat ini (Santoso, 2005).

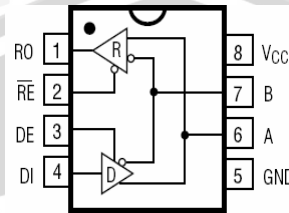
2.2 Device Network Interface Card (NIC) RS-485

RS-485 merupakan pengembangan komunikasi serial untuk menjawab kelemahan komunikasi serial standar RS-232 yang terdapat pada *PC*. RS-232 memiliki kelemahan pada jumlah percabangan meskipun dapat mencapai jarak yang cukup jauh. RS-232 hanya dapat berkomunikasi satu titik ke satu titik yang lain (*point to point*). RS-485 dengan penyempurnaan perancangan dapat dipecah hingga mencapai 32 titik (*multipoint*) dengan jarak relatif lebih jauh dibanding RS-232 yang dapat mencapai jarak 1,2 km (Axelson, 1999).

Pengiriman data pada komunikasi serial RS-485 bersifat *half duplex*. Pengiriman data secara *half duplex* adalah proses pengiriman data yang tidak dapat dilakukan secara dua arah (mengirim dan menerima data) dalam satu waktu tetapi harus bergantian. Pengiriman data secara bergantian mengakibatkan adanya proses menunggu hingga proses pengiriman data yang berlangsung telah selesai.

NIC RS-485 memiliki delapan buah pin yang memiliki fungsi berbeda-beda. Pin nomor satu (*Receiver Output*) merupakan pin yang akan menerima data pada perangkat komunikasi lain. Pin nomor dua (*Receiver Output Enable*) merupakan pin yang mengatur kinerja dari pin RO. Pin RO akan berfungsi apabila pin $\overline{RE}$ memiliki logika rendah atau bernilai nol. Pin nomor tiga (*Driver Output Enable*) merupakan pin yang berfungsi untuk mengatur kinerja dari pin nomor empat (*Driver Input*) yang akan mengirimkan data menuju perangkat lain pada saat pin nomor tiga berlogika tinggi. Pin nomor lima memiliki fungsi sebagai *ground* dari IC RS-485. Pin nomor enam (A) dan tujuh (B) sebagai jalur data yang dalam bekerjanya bergantung pada pin $\overline{RE}$ dan DE. Apabila pin $\overline{RE}$ berlogika rendah maka proses penerimaan data akan berlangsung dan bila pin DE berlogika tinggi maka proses pengiriman data akan

berlangsung. Pin A dan B dapat bekerja bila memiliki selisih tegangan antara A dan B sebesar 200mV.



Gambar 2.2 Blok diagram NIC RS-485 dengan delapan pin yang memiliki fungsi berbeda-beda.

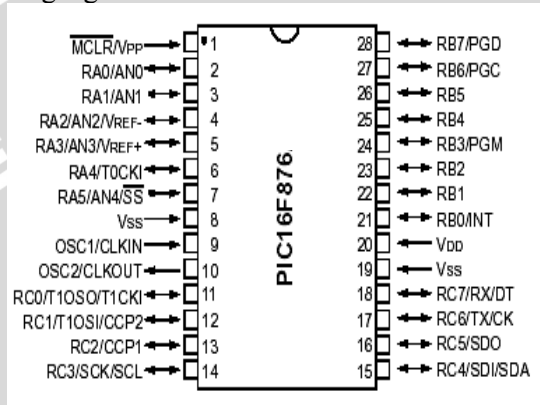
Bila antara jalur A dan B memiliki selisih tegangan kurang dari 200mV maka nilai ini tidak didefinisikan atau RS-485 tidak dapat bekerja sebagaimana mestinya. Pin nomor delapan (Vcc) memiliki fungsi sebagai masukan tegangan sebesar lima volt.

2.3 Mikrokontroler PIC16F876

Mikrokontroler merupakan suatu IC yang memiliki kemampuan proses berfikir berdasarkan program yang telah diberikan atau diisikan pada mikrokontroler tersebut. Mikrokontroler memiliki beberapa bagian untuk mendukung proses berfikir seperti CPU, ROM, RAM dan I/O. CPU (*Central Processing Unit*) berfungsi untuk membaca program yang tersimpan di dalam ROM dan melaksanakannya. ROM (*Read Only Memory*) merupakan memori yang hanya dapat dibaca saja untuk menyimpan program dari mikrokontroler itu sendiri. RAM (*Random Acces Memory*) merupakan memori yang dapat dibaca dan ditulis atau diisi berulang kali oleh pemrogram. I/O (*Input/Output*) merupakan bagian penghubung antara mikrokontroler dengan perangkat luar agar dapat berkomunikasi antara keduanya (Malik, 2003).

Mikrokontroler memiliki beberapa fitur tambahan untuk mendukung proses berfikir dari mikrkontroler. Salah satu kompoen tambahan tersebut adalah ADC (*Analog To Digital Converter*). ADC berfungsi untuk mengubah nilai analog yang masuk pada mikrokontroler menjadi nilai digital. Dengan pengubahan nilai analog menjadi nilai digital maka segala bentuk masukan analog dari luar mikrokontroler dapat diterjemahkan oleh mikrokontroler maupun perangkat lain yang membutuhkan masukan berupa nilai digital.

Mikrokontroler PIC16F876 merupakan mikrokontroler yang diproduksi oleh Microchip Inc. Mikrokontroler jenis ini memiliki 8kByte *flash memory*, tiga *port* I/O yaitu port A, B dan C, masukan analog sebanyak lima *channel* yang telah dilengkapi dengan ADC sebesar 10 bit pada port A. Mikrokontroler PIC16F876 membutuhkan tegangan sebesar 5 volt.



Gambar 2.3 Susunan pin mikrokontroler PIC16F876 dengan fungsi yang berbeda bergantung pengaturan pada saat inisialisasi.

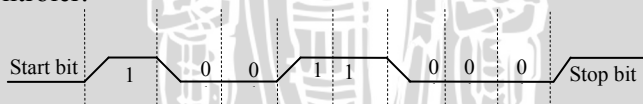
Mikrokontroler PIC16F876 memiliki ukuran fisik yang kecil dengan jumlah kaki 28 pin. Masing-masing pin memiliki fungsi yang berbeda dan beberapa pin memiliki fungsi ganda. Pengaturan fungsi pin dilakukan pada saat inisialisasi mikrokontroler. Pin *MCLR* atau pin nomor satu berfungsi untuk mengembalikan kondisi mikrokontroler pada keadaan semula (*reset*) pada saat diberi logika rendah. Pin RA0 hingga RA5 adalah *port* A yang berfungsi sebagai ADC pada saat menerima masukan nilai analog. Pin Vss sebagai masukan *ground* pada mikrokontroler. Pin nomor sembilan dan sepuluh sebagai masukan pengatur *clock* berupa kristal yang dibutuhkan oleh mikrokontroler. Pin RC0 hingga RC7 merupakan *port* C yang berfungsi sebagai *input* atau *output* dari mikrokontroler. Pada pin nomor tujuh belas dan delapan belas memiliki fungsi ganda sebagai jalur komunikasi serial antara mikrokontroler dengan perangkat luar. Pin nomor tujuh belas dapat berfungsi sebagai jalur pengirim data (TX) dari mikrokontroler dan pin nomor delapan belas

sebagai jalur penerima data (RX) bila fungsi khusus tersebut diaktifkan pada saat inisialisasi mikrokontroler. Pin Vss memiliki fungsi sebagai jalur masukan sinyal *ground* seperti pada pin nomor delapan. Pin Vdd sebagai pin sumber tegangan sebesar 5 volt agar mikrokontroler dapat bekerja. Pin RB1 hingga RB7 memiliki fungsi sebagai jalur *input* atau *output* dari mikrokontroler bergantung pada saat inisialisasi.

2.4 Port Serial Komputer

Komunikasi serial adalah proses pengiriman data secara serial atau data dikirim satu per satu secara berurutan. Komunikasi serial memiliki kelebihan pada jangkauan panjang kabel yang lebih jauh dibandingkan dengan paralel. Serial port mengirimkan logika 1 dengan tegangan +3 volt hingga +25 volt dan logika 0 sebagai +3 volt hingga +25 volt.

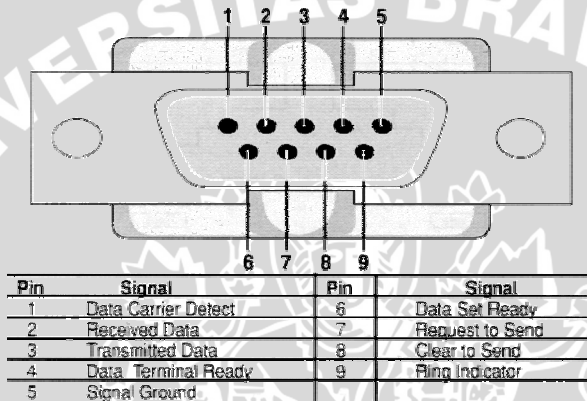
Komunikasi serial port bersifat asinkron. Komunikasi serial asinkron pada saat pengiriman data harus diawali dengan *start* bit dan diakhiri dengan stop bit. Sinyal *clock* yang merupakan *baud rate* dari komunikasi data dibangkitkan oleh penerima maupun pengirim data dengan frekuensi yang sama. Penerima akan mendeteksi adanya *start* bit sebagai awal pengiriman data dan setelah data dikirim oleh pengirim, penerima akan menunggu stop bit sebagai tanda bahwa data telah selesai dikirim. Pada aplikasinya proses komunikasi asinkron selalu digunakan untuk mengakses komponen-komponen yang memiliki fasilitas *UART (Universal Asynchronous Receiver/Transmitter)* seperti port serial PC maupun port serial mikrokontroler.



Gambar 2.4 Pengiriman data secara asinkron yang diawali *start* bit dan diakhiri stop bit.

Perangkat keras pada komunikasi serial port dibagi menjadi dua kelompok yaitu *Data Communication Equipment (DCE)* dan *Data Terminal Equipment (DTE)*. Contoh DCE adalah modem sedangkan DTE adalah terminal serial di komputer. Konektor port serial terdiri dari dua jenis yaitu konektor 25 pin atau biasa disebut dengan DB25 dan 9 pin atau biasa disebut dengan DB9 (Budiharto, 2004).

Pin pada DB9 memiliki fungsi yang berbeda-beda seperti pada gambar 2.5. Pin yang digunakan untuk komunikasi serial RS-485 antara PC dengan mikrokontroler adalah pin nomor 2, 3 dan 7. Pin *received data* berfungsi sebagai jalur penerima data yang dikirim oleh mikrokontroler. Pin *transmitted data* berfungsi sebagai jalur pengirim data menuju mikrokontroler dan pin *request to send* sebagai jalur penanda kesiapan pengiriman data dari PC menuju mikrokontroler.



Gambar 2.5 Pin pada DB9 dengan fungsi masing-masing pin yang berbeda. Pada komunikasi serial RS-485 dipergunakan pin 2, 3 dan 7 untuk mendukung komunikasi antara PC dengan mikrokontroler.



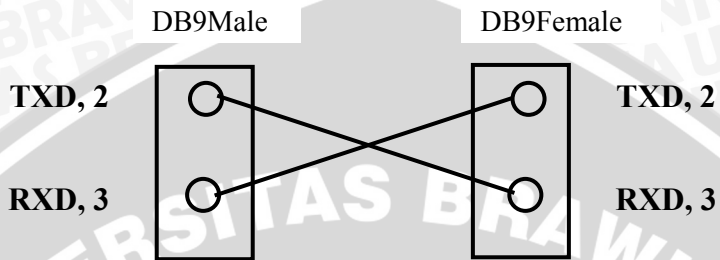
Gambar 2.6 DB9Male



Gambar 2.7 DB9Female

Pin DB9Female (betina) dengan DB9Male (jantan) memiliki kesamaan fungsi dari masing-masing pin tetapi perbedaan terjadi pada saat pemasangan keduanya. Fungsi-fungsi pin tersebut dipasang saling bersilangan seperti pin *transmit data* dari DB9Female dihubungkan dengan pin *received data* dari DB9Male dan pin

received data (RxD) dari DB9Female dihubungkan dengan pin *transmit data (TxD)* dari DB9Male.



Gambar 2.8 Pemasangan DB9 Female dan DB9 Male secara bersilangan.

2.5 Sensor Suhu

Temperatur atau suhu dalam kehidupan sehari-hari sering dianggap sebagai ukuran panas atau dinginnya suatu benda. Namun secara mikroskopis temperatur merupakan energi kinetik internal rata-rata suatu benda (Tipler, 1998).

Alat pengukur temperatur biasa disebut dengan termometer. Termometer dalam cara kerjanya bergantung pada sifat materi yang digunakan sebagai penunjuk adanya perubahan temperatur. Pada masa awal penemuan termometer oleh Galileo dimanfaatkan pemuaian gas sebagai indikator perubahan temperatur. Namun saat ini ada bermacam termometer dari bahan logam cair hingga memanfaatkan sifat kelistrikan suatu bahan (Giancoli, 2001).

Salah satu jenis sensor suhu yang umum dipakai adalah LM35. Sensor LM35 memiliki beberapa macam bentuk, salah satunya memiliki tiga kaki dengan bentuk menyerupai transistor. Sensor ini menghasilkan tegangan yang sebanding dengan perubahan temperatur tiap derajat Celcius. Sensor LM35 dapat menghasilkan tegangan sebesar 250mV pada suhu 25°C dan 1000mV pada suhu 100°C atau bisa dikatakan sensor LM35 memiliki perubahan sebesar 10mV untuk perubahan setiap satu derajat Celcius (10mV/°C) (National Semiconductor, 1995).

2.6 Sensor Kelembaban

Kelembaban relatif merupakan persentase antara tekanan parsial dari H₂O terhadap tekanan uap jenuh dari H₂O. Tekanan parsial merupakan tekanan yang dimiliki salah satu komponen penyusun

udara terhadap tekanan total seluruh komponen penyusun udara. Tekanan uap jenuh merupakan tekanan yang dimiliki komponen udara pada saat menjadi uap jenuh. Persamaan kelembaban relatif dituliskan dalam bentuk persamaan 2.1 (Giancoli, 2001).

$$\text{Kelembaban Relatif} = \frac{\text{tekanan parsial H}_2\text{O}}{\text{tekanan uap jenuh H}_2\text{O}} \times 100\% \quad (2.1)$$

Sensor kelembaban HS15P merupakan sensor kelembaban tahan air. Sensor ini bekerja pada *range* kelembaban 20%-100%. Sensor ini dalam bekerjanya membutuhkan tegangan sebesar 1VAC dengan frekuensi sebesar 1kHz. Sensor HS15P bekerja bergantung pada perubahan nilai resistansi terhadap lingkungan. Nilai resistansi sensor akan mengalami perubahan bertambah besar pada saat kelembaban lingkungan makin rendah dan resistansi akan berkurang pada saat kelembaban lingkungan makin tinggi (GE Sensing, 2005).



Gambar 2.9 Sensor kelembaban HS15P membutuhkan tegangan 1VAC dan frekuensi 1kHz.

BAB III METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

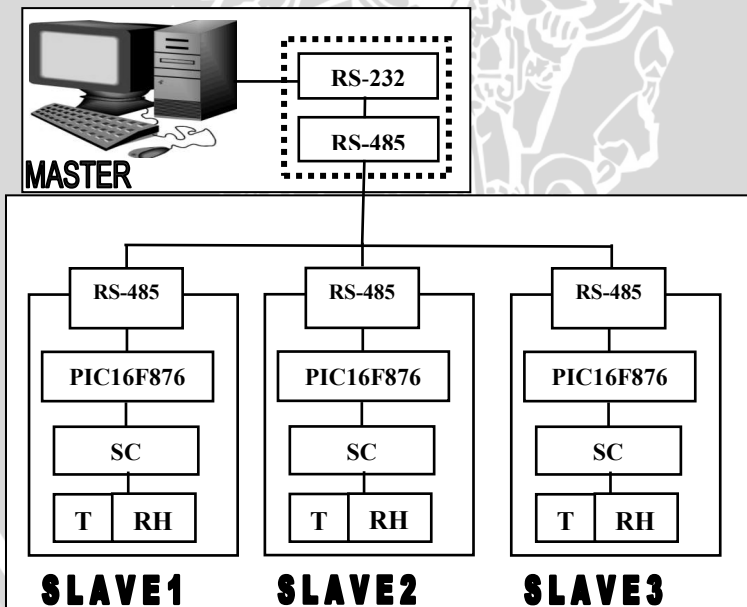
Penelitian dilakukan pada bulan Januari 2007 - Nopember 2007 di laboratorium Instrumentasi dan Pengukuran Jurusan Fisika FMIPA Universitas Brawijaya Malang.

3.2. Tahapan Pengerjaan

Tahapan pengerjaan penelitian ini adalah sebagai berikut:

- i. Studi literatur
- ii. Perancangan dan pembuatan sistem
- iii. Pengujian dan pengambilan data
- iv. Analisa hasil
- v. Pelaporan

3.3. Perancangan Perangkat Keras



Gambar 3.1 Perancangan perangkat keras meliputi *slave*, RS-485 To RS-232 Converter dan PC.

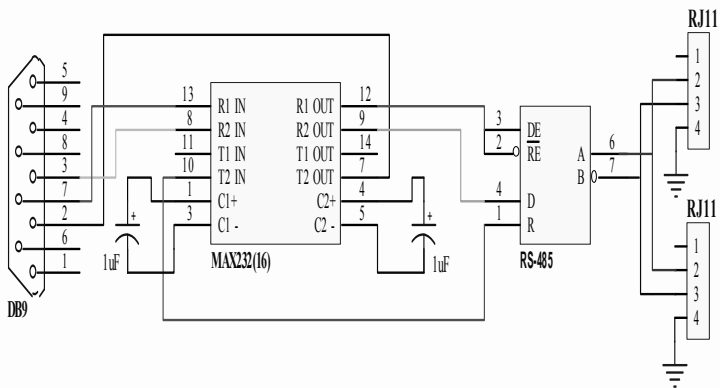
Blok diagram pada gambar 3.1 menggambarkan rancangan yang dibuat dalam penelitian. Dua sensor yaitu sensor suhu (T) dan sensor kelembaban (RH) akan mendeteksi adanya perubahan suhu dan kelembaban yang terjadi pada suatu tempat. Nilai perubahan yang diterima oleh kedua sensor diolah oleh *signal conditioner (SC)* atau pengondisi sinyal berupa penguat maupun pengubah tegangan AC ke DC (*ACDC Converter*). Adanya pengondisi sinyal bertujuan untuk memperbaiki kualitas dari keluaran yang dihasilkan oleh sensor. Sensor tanpa dilewatkan *signal conditioner* mengakibatkan perubahan nilai yang dihasilkan tidak dapat diolah oleh ADC. Dari masing-masing pengondisi sinyal diteruskan menuju mikrokontroler yang telah dilengkapi dengan *ADC*. Penggunaan *ADC* bertujuan merubah besaran analog yang merupakan keluaran dari kedua sensor menjadi nilai digital agar dapat diterima dan diolah oleh mikrokontroler sebagai pemroses seluruh bagian dari sistem pada *slave*. Dari mikrokontroler menuju RS-485 sebagai media antar muka antara *slave* yang satu dengan *slave* yang lain untuk diteruskan menuju *PC*. Sebelum menuju *PC* terlebih dahulu dilakukan pengubahan komunikasi *interface* dari RS-485 menjadi RS-232 agar kompatibel dengan komunikasi *interface* yang ada pada *PC*. Data diolah lebih lanjut oleh *PC* sehingga dapat disimpan dan ditampilkan.

3.3.1 Perancangan Modul Master

Perancangan yang dilakukan pada bagian ini berupa perancangan *RS-485 To RS-232 Converter* sebagai pendukung dan pengubah komunikasi serial RS-485 dari *slave* menjadi RS-232 agar sesuai dengan komunikasi serial yang terdapat pada *master* dalam hal ini *PC*.

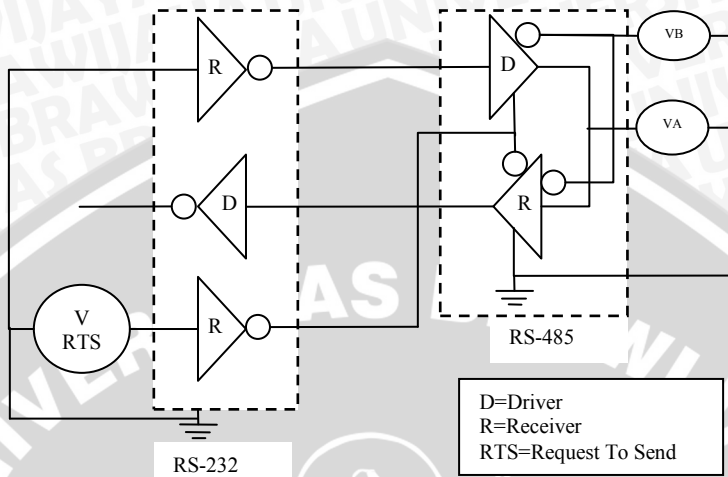
Rangkaian pada gambar 3.2 berfungsi sebagai pengubah komunikasi RS-485 dari mikrokontroler menjadi komunikasi serial RS-232 pada *PC*. Bagian awal rangkaian terdapat RJ11 dengan pin nomor 2 dihubungkan pada pin 7 RS-485 dan pin nomor 3 dihubungkan dengan pin nomor 6 RS-485. Dari RS-485 pin RE dan DE dihubungkan dengan pin R1Out (12) RS-232, pin RO RS-485 dengan pin R2Out (10) RS-232 dan pin DI RS-485 dihubungkan dengan pin R2IN (9) RS232. Dari RS-232 pin T2IN (7) dihubungkan dengan pin *Receive Data* (2) DB9F, pin R2Out (8) dihubungkan

dengan pin *Transmit Data* (3) DB9F dan pin RTS(7) dari DB9 dihubungkan dengan pin R1/IN (13) RS-232.



Gambar 3.2 Rangkaian RS-485 To RS-232 Converter memanfaatkan pin nomor 2 dan 3 pada DB9Female sebagai jalur pengirim dan penerima data serta pin nomor 7 sebagai pengatur kinerja NIC RS-485.

Sebuah sistem komunikasi RS-485 membutuhkan pengendali yang dapat mengatur koneksi jalur pengiriman data. Pengendali ini akan mengatur tersambung atau terputusnya jalur komunikasi pada RS-485. Pengendali koneksi dapat dilakukan dengan memanfaatkan sinyal *request to send (RTS)* pada port serial. Jalur RTS yang terhubung pada pin *driver enable* memberikan logika tinggi agar RS-485 dapat bekerja. Sedangkan pada logika rendah akan mengakibatkan kondisi *tristate* sehingga sebuah jalur transmisi akan terputus dan membolehkan jalur lain untuk mengirimkan data meskipun menggunakan jalur kabel yang sama. Sinyal RTS digunakan untuk mengeset logika tinggi pada saat data sebelum dikirim dan mengeset pada logika rendah sesudah data dikirim. Pengaturan logika tinggi dan rendah dari jalur RTS diatur melalui *software*.



Gambar 3.3 Blok diagram RS-232 To RS-485 *Converter* yang diatur dari *software* dengan memanfaatkan jalur RTS pada port serial DB9.

Nilai logika yang dikirimkan oleh *master* melalui port serial akan selalu dibalik oleh RS-232 karena jalur pengiriman data melalui *inverter* (pembalik) pada RS-232. *Inverter* akan membalik nilai logika tinggi yang dikirim oleh *master* menjadi logika rendah dan nilai logika rendah menjadi logika tinggi. Perancangan *software* pada *master* untuk membuka koneksi transmisi data dikirimkan pertama kali logika rendah agar RS-485 mengaktifkan *driver enable* yang berlogika tinggi dan segera mengirimkan logika tinggi untuk mengaktifkan *receiver enable* yang berlogika rendah sesudah data dikirim oleh *master*.

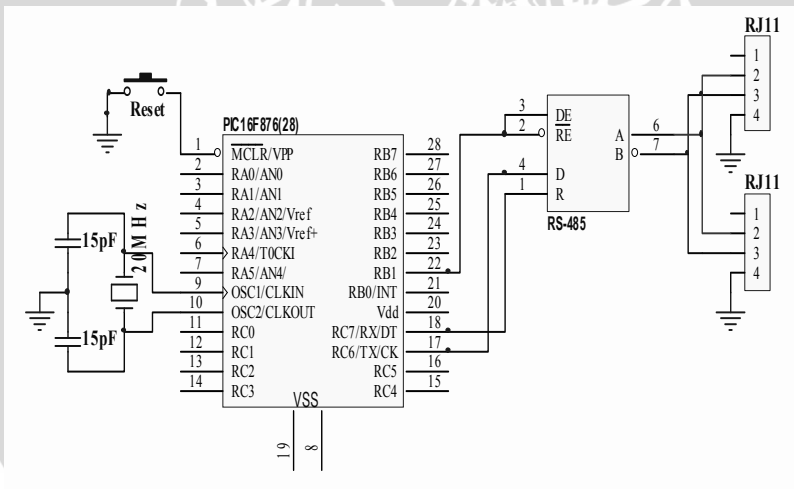
Master secara umum memberikan perintah permintaan pengiriman data dari tiap *slave*. Agar data dari tiap *slave* tidak tertukar, *master* akan mengirimkan karakter kunci yang berbeda pada tiap *slave*. Pada *master* telah ditempatkan kolom-kolom yang akan menyimpan data hasil pengukuran dari tiap *slave*.

3.3.2 Perancangan Modul *Slave*

Perancangan pada modul *slave* meliputi dua hal, yaitu perancangan *hardware* dan *software*. Perancangan *hardware* yang diikuti dengan perancangan *software* bertujuan untuk mendukung kinerja dari *slave* sebagai pengirim data hasil pengukuran.

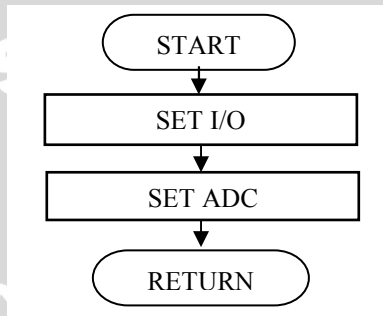
Modul *slave* terdiri dari sensor, mikrokontroler dan *NIC* RS-485. Sensor berfungsi sebagai sarana pendeteksi adanya perubahan besaran fisis pada suatu tempat. Mikrokontroler berfungsi sebagai pengubah nilai analog yang dihasilkan sensor menjadi nilai digital juga sebagai pengendali komunikasi baik dari *slave* ke *master* maupun dari *master* menuju *slave*. *NIC* RS-485 sebagai penunjang utama komunikasi antara *master* dengan *slave*. *NIC* RS485 diatur oleh mikrokontroler dalam proses transmisi data baik dari master menuju *slave* maupun dari *slave* menuju *master*.

Mikrokontroler yang digunakan pada modul ini adalah mikrokontroler PIC16F876. Sistem ini memanfaatkan *port* A sebagai masukan analog yaitu RA0 dan RA1. Kedua *channel* tersebut secara berurutan menjadi masukan dari sensor suhu dan kelembaban. Sebagai pengontrol aktifasi dari RS-485 digunakan *port* B yaitu RB1. Pin RB1 terhubung dengan RS-485 pada pin RE dan DE, sedangkan *port* RX mikrokontroler terhubung pada pin RO dan pin TX mikrokontroler pada pin DI. Keluaran dari RS-485 terletak pada pin A dan B yang dihubungkan dengan RJ11 kaki nomor 2 dan nomor 3.



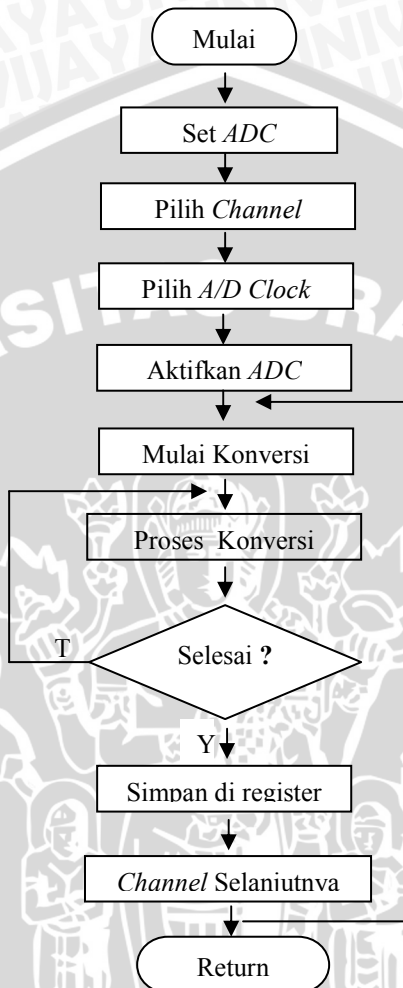
Gambar 3.4 Rangkaian modul *Slave* berupa mikrokontroler (telah dilengkapi *NIC* RS-485) sebagai pengolah dan pengirim data pengukuran.

Tahap awal sesudah perancangan *hardware* adalah menginisialisasi mikrokontroler agar dapat bekerja dan berfungsi sebagai pengubah nilai analog menjadi digital maupun sebagai pengendali komunikasi antara *slave* dengan *master*. Inisialisasi meliputi inisialisasi *baudrate* yang digunakan, jumlah masukan analog, port keluaran pengatur komunikasi RS-485, dan jalur pengiriman data.



Gambar 3.5 Prosedur inisialisasi pada mikrokontroler.

Prosedur pada gambar 3.5 diawali dengan pengesetan I/O pada mikrokontroler, yaitu pengesetan dua masukan analog pada port A, port B sebagai keluaran serta pengesetan fungsi RX/TX untuk proses pengiriman maupun penerimaan data. Kemudian diikuti dengan pengesetan modul *ADC* pada mikrokontroler.



Gambar 3.6 Prosedur konversi pada mikrokontroler

Prosedur pada gambar 3.6 merupakan proses konversi dengan terlebih dahulu melakukan pengaturan konfigurasi *ADC*. Konfigurasi *ADC* disini dengan menjadikan RA0 hingga RA5 sebagai masukan nilai analog yang dihasilkan oleh sensor. Konfigurasi masukan analog ini sekaligus menjadikan *port A* sebagai *port* masukan. Kemudian dilanjutkan dengan mengatur berapa *channel ADC* yang diaktifkan. Kemudian dilanjutkan dengan pengiriman data melalui *port* keluaran, disamping itu juga mengatur nilai *clock ADC* serta

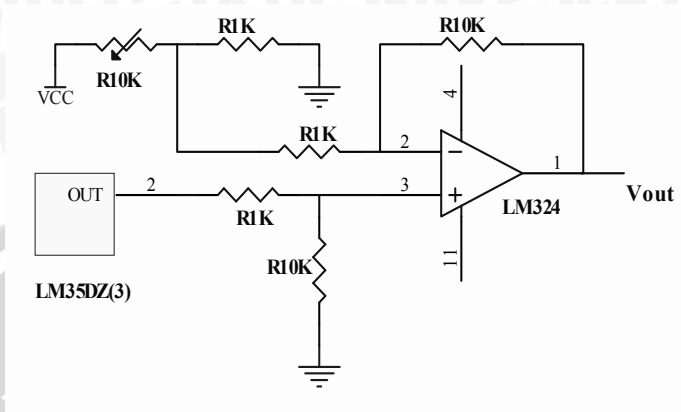
aktifasi dari *ADC*. Sesudah konfigurasi *ADC* dilakukan maka ketika ada masukan berupa nilai analog pada *port* yang telah dikonfigurasi, proses konversi dapat mulai dilakukan setelah ada perintah untuk memulai konversi nilai analog menjadi nilai digital melalui *software* yang telah diset pada register *ADCON0*. Proses ini dijalankan pada tiap *channel ADC* hingga proses konversi selesai, bila proses konversi tersebut sudah selesai akan disimpan pada register yang telah ditentukan dan mikrokontroler akan melanjutkan proses konversi pada *channel* berikutnya dengan memulai proses awal konversi. Proses konversi akan terus berlanjut pada *channel* berikutnya dan sesudah pada *channel* terakhir akan kembali pada *channel* pertama.

Proses pengiriman data pada *slave* diawali dengan adanya pengiriman alamat dari master atau *PC*. Pengiriman alamat ini merupakan pemberian logika rendah menuju RS-485 pada jalur A. Karakter kunci yang dikirimkan dibedakan menjadi dua fungsi yaitu sebagai perintah untuk memulai proses konversi pada *ADC* mikrokontroler dan sebagai alamat dari tiap *slave*. Apabila karakter yang diterima adalah perintah konversi maka mikrokontroler akan memulai proses konversi. Apabila karakter kunci yang diterima berupa alamat dan sesuai dengan alamat *slave*, mikrokontroler akan memberikan nilai logika tinggi pada port B1. Pemberian logika tinggi ini akan mengakibatkan jalur B pada RS-485 aktif tinggi sehingga hasil konversi akan dikirimkan menuju *master* dalam hal ini pada *PC*.

3.3.3 Perancangan Rangkaian Sensor

a. Rancangan Sensor Suhu

Rangkaian pada gambar 3.7 berfungsi untuk memperkuat tegangan hasil dari mendeteksi suhu lingkungan yang diukur. Nilai keluaran sensor tanpa pengkondisi sinyal untuk tiap derajat celsius bernilai 10mvolt sehingga dibuat penguat 10 kali agar nilai keluaran tersebut dapat dibaca oleh mikrokontroler untuk dikonversi menjadi nilai digital.



Gambar 3.7 Rangkaian sensor suhu

Penguat yang digunakan adalah penguat diferensial dengan tujuan untuk membatasi *range* suhu yang dapat dideteksi oleh sensor. Pada masukan *inverting* dari penguat diferensial tegangan masukan yang diterima adalah keluaran dari rangkaian pembagi tegangan. Rangkaian pembagi tegangan akan membatasi nilai suhu yang bernilai kurang dari 15 °C sehingga pada saat suhu sensor sebesar 15 °C tegangan yang terbaca sebesar 0 volt. Persamaan pembagi tegangan secara umum biasa dituliskan dalam bentuk persamaan 3.1.

$$v_{out} = \frac{R_2}{R_2 + R_1} \times V_{input} \quad (3.1)$$

Nilai R_2 sebesar $1k\Omega$ dan R_1 berupa trimpot $10k\Omega$ dengan tegangan input sebesar 5 volt. Bila nilai R_2 dan R_1 dimasukkan persamaan 3.1 maka

$$v_{out} = \frac{1k\Omega}{1k\Omega + 10k\Omega} \times 5volt$$

$$v_{out} = 450mvolt$$

Suhu yang dibatasi sebesar 15°C sehingga pembatas tegangan tersebut diatur agar keluaran rangkaian pembagi tegangan sebesar 150 mvolt dengan memutar trimpot. Tegangan masukan *non inverting* pada rangkaian penguat diferensial adalah tegangan keluaran yang dihasilkan oleh sensor LM35.

Tegangan keluaran pada penguat diferensial biasa dituliskan dalam bentuk persamaan 3.2.

$$v_{out} = \frac{R_2}{R_1} (v_1 - v_2) \quad (3.2)$$

Nilai penguatan sebesar 10 kali diperoleh dari nilai perbandingan antara R_2 sebesar $10k\Omega$ dengan R_1 sebesar $1k\Omega$. Sehingga tegangan keluaran dari penguat diferensial menjadi

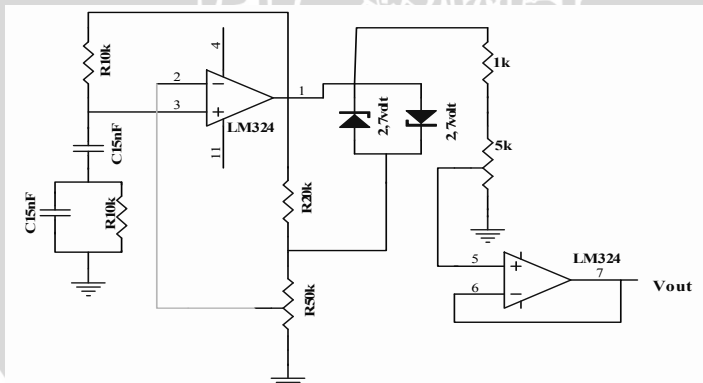
$$v_{out} = \frac{10k\Omega}{1k\Omega} (v_{sensor} - 150mvolt)$$

b. Rangkaian Sensor Kelembaban

Sensor HS15P dalam bekerjanya membutuhkan tegangan 1VAC dengan frekuensi 1kHz, nilai ini dapat dihasilkan dari rangkaian osilator *Wien* dan rangkaian pembagi tegangan yang ditempatkan pada keluaran osilator. Rangkaian osilator *Wien*, terdiri dari sepasang komponen resistor dan kapasitor yang tersusun seri dan paralel yang diteruskan pada op-amp. Frekuensi sebesar 1kHz dapat diperoleh secara matematis dari persamaan 3.3.

$$f = \frac{1}{2\pi RC} \quad (3.3)$$

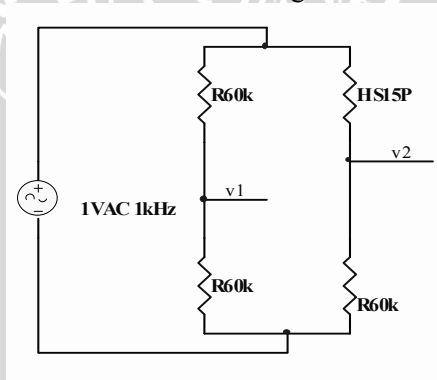
Dengan menentukan nilai resistor sebesar $10k\Omega$ maka diperlukan kapasitor 15nF.



Gambar 3.8 Osilator *Wien* berfrekuensi 1kHz dengan keluaran tegangan sesudah rangkaian pembagi tegangan sebesar 1 VAC.

Tegangan keluaran dari osilator menjadi masukan rangkaian pembagi tegangan agar diperoleh tegangan sebesar 1VAC. Rangkaian pembagi tegangan memiliki nilai R_2 sebesar $1k\Omega$ dan R_1 sebesar $5k\Omega$, dengan mengubah-ubah nilai trimpot maka nilai tegangan 1VAC dapat diperoleh dari rangkaian tersebut. Tegangan keluaran dari rangkaian pembagi tegangan menjadi masukan rangkaian pengikut tegangan untuk menjaga stabilitas tegangan yang dihasilkan oleh rangkaian pembagi tegangan tanpa melakukan penguatan tegangan.

Sensor HS15P dalam penggunaannya memanfaatkan rangkaian jembatan *wheatstone* seperti pada gambar 3.9 agar dapat menghasilkan tegangan keluaran. Pada rangkaian ini terdiri dari empat nilai tahanan, dimana salah satu dari empat nilai tahanan tersebut digantikan dengan sensor HS15P. Tiga resistor yang terpasang merupakan resistor referensi berdasarkan pada *datasheet* dari sensor yang memiliki *range* resistansi dari $60k\Omega$ hingga $80k\Omega$. Perubahan nilai pada sensor HS15P ini akan mempengaruhi nilai keluaran pada rangkaian jembatan *wheatstone*, sebagaimana bila dilakukan pengubahan nilai resistor pada rangkaian jembatan, maka akan ada perbedaan nilai keluaran dari rangkaian tersebut.



Gambar 3.9 Rangkaian jembatan *WheatStone* dengan sensor HS15P sebagai nilai hambatan yang berubah. Nilai tegangan yang dihasilkan merupakan selisih dari $v1-v2$.

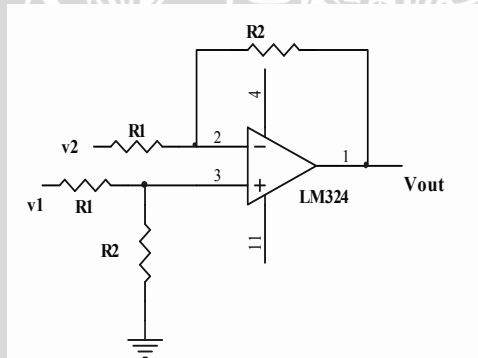
Untuk memperoleh keluaran dari sensor HS15P, sensor diletakkan pada salah satu bagian dari jembatan *wheatstone* tersebut dengan masukan sebesar 1VAC 1kHz sesuai *datasheet* untuk menghasilkan tegangan pada setiap perubahan kelembaban udara. Perubahan

kelembaban udara akan menyebabkan perubahan resistansi dari sensor, perubahan resistansi akan menyebabkan jembatan tidak setimbang sehingga menghasilkan tegangan pada titik *output* dari rangkaian jembatan yang selanjutnya tegangan tersebut akan diperkuat oleh penguat diferensial.

Nilai tegangan yang dihasilkan dari rangkaian pada gambar 3.9 dapat dihitung secara matematis dengan menerapkan persamaan selisih pembagi tegangan seperti pada persamaan 3.4.

$$V_{out} = \left[\left(\frac{R_3}{R_1 + R_3} \right) - \left(\frac{R_4}{R_2 + R_4} \right) \right] \times V_{in} \quad (3.4)$$

Dimana nilai R_1 , R_2 , R_3 merupakan nilai resistor yang ditentukan berdasar *datasheet* dari sensor HS15P. Sedangkan R_4 merupakan nilai hambatan yang dimiliki oleh sensor yang nilainya dapat berubah-ubah sesuai kondisi lingkungan yang diukur. Dan V_{in} merupakan tegangan yang telah ditentukan oleh perusahaan yaitu sebesar 1VAC.

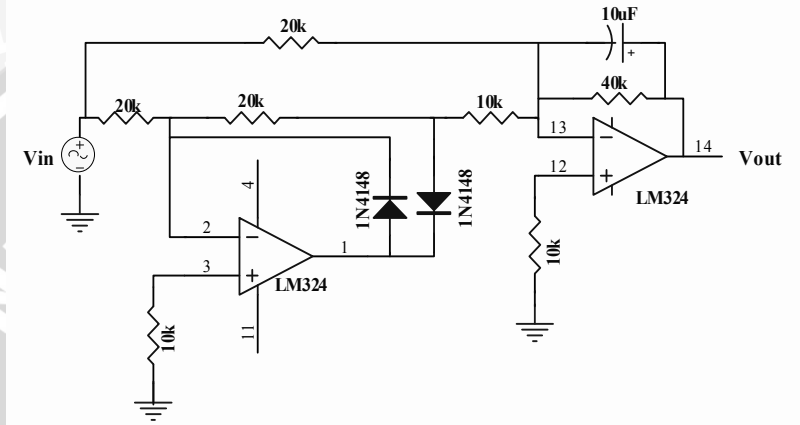


Gambar 3.10 Rangkaian penguat diferensial

Penguat diferensial pada gambar 3.10 menghasilkan nilai penguatan sebesar 10 kali dengan nilai $R1=100k\Omega$ dan $R2=10k\Omega$. Besar penguatan diferensial yang diperoleh apabila kita masukkan dalam persamaan 3.2 dengan nilai V_{in} selisih dari tegangan pada masukan inverting dan non inverting. Rangkaian ini dipergunakan dengan pertimbangan nilai keluaran tegangan dari rangkaian jembatan *wheatstone* terlampau kecil untuk dibaca pada perangkat mikrokontroler dan dibutuhkan penguatan sebesar 10 kali. Hasil

penguatan tersebut kemudian diteruskan menuju rangkaian penyearah untuk mengubah tegangan yang dihasilkan.

Rangkaian penyearah pada gambar 3.11 berfungsi mengubah tegangan keluaran dari penguat yang berupa tegangan AC menjadi tegangan DC.



Gambar 3.11 Rangkaian penyearah tegangan AC menjadi DC.

Pengubahan bentuk tegangan bertujuan agar tegangan keluaran yang dihasilkan sensor dapat dibaca dan diolah oleh mikrokontroler. Tegangan keluaran sensor yang berupa tegangan AC harus disearahkan menjadi tegangan DC. Dengan pengubahan bentuk tegangan dari tegangan AC menjadi tegangan DC, mikrokontroler dapat mengubah tegangan keluaran sensor menjadi nilai digital.

Tegangan AC negatif yang masuk pada masukan inverting dari op-amp pertama akan dibalik menjadi tegangan positif. Tegangan positif dilewatkan kembali melalui dioda pertama menuju masukan inverting pada op-amp pertama dan dioda kedua tidak bekerja. Hasil keluaran kedua dari op-amp pertama menjadi tegangan negatif dan diteruskan oleh dioda kedua menuju masukan inverting op-amp kedua menjadi tegangan positif.

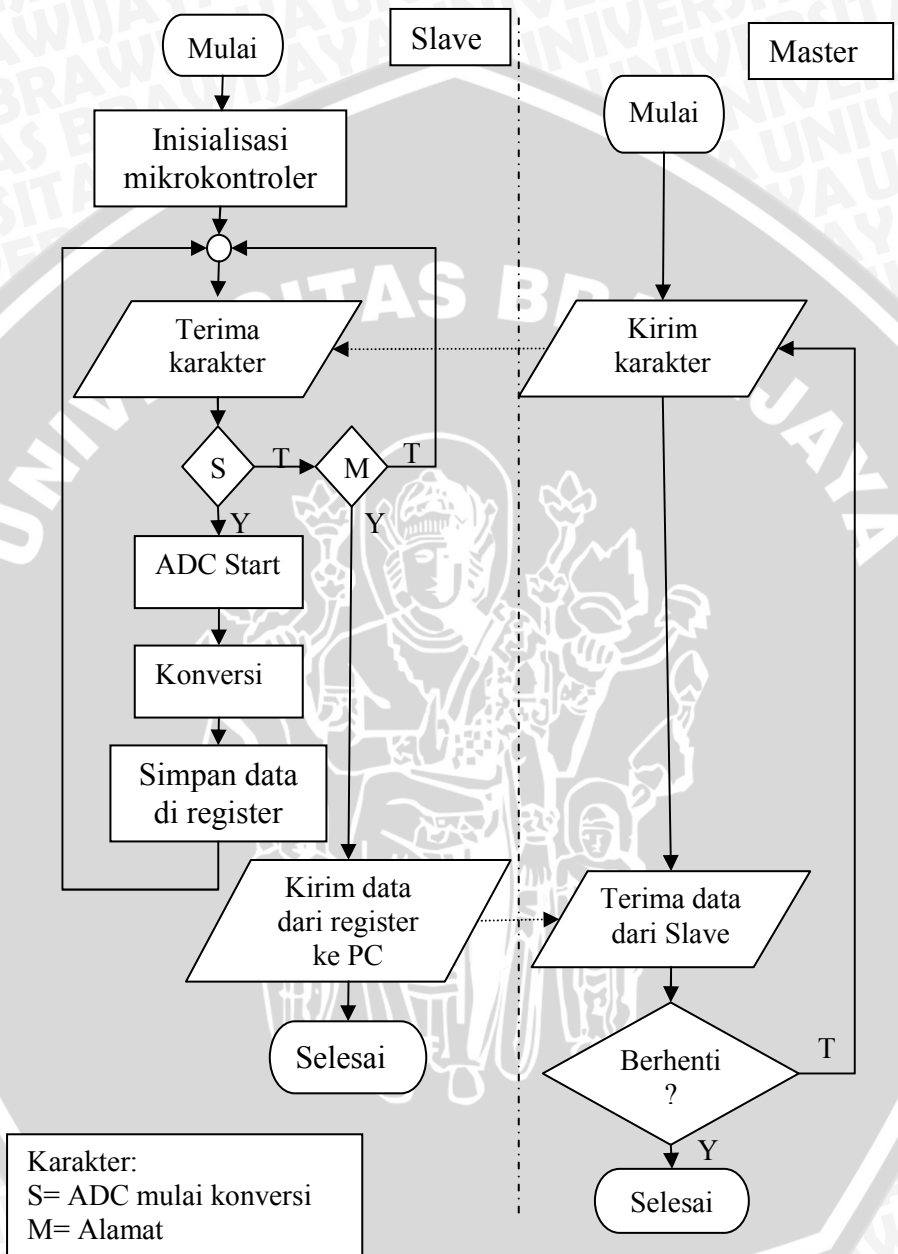
Tegangan AC positif yang masuk pada masukan inverting op-mp pertama mempunyai keluaran berupa tegangan negatif. Keluaran berupa tegangan negatif dari op-mp pertama langsung diteruskan oleh dioda kedua menuju op-amp kedua menuju masukan inverting op-amp kedua. Masukan negatif dari op-amp pertama berubah menjadi tegangan positif sesudah keluar dari op-amp kedua.

3.3.4 Perancangan Protokol Komunikasi

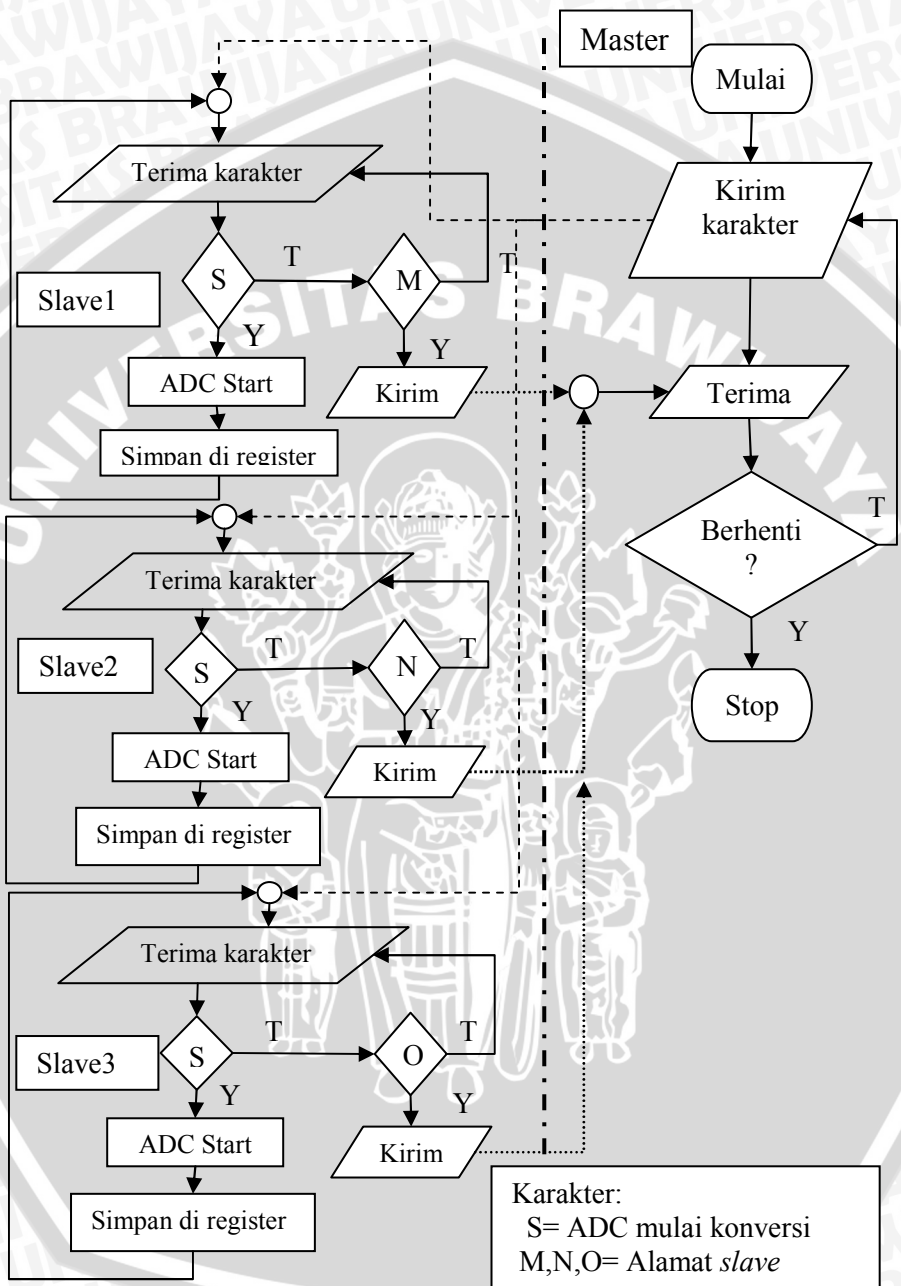
Pada perancangan protokol komunikasi dibahas jalannya komunikasi antara *master* dengan *slave* dan proses transfer data dari *slave* ke *master*.

Transfer data dapat dilakukan setelah dilakukan penyesuaian *baudrate* dan *port serial* yang digunakan. *Baudrate* yang digunakan 9600bps sesuai dengan yang diset pada mikrokontroler untuk mendukung proses transfer data.

Prosedur pada gambar 3.12 sebagai contoh transfer data pada *slave* 1 dimulai dengan perintah 'S' yang dikirim dari *master* menuju *slave* untuk mengaktifkan ADC pada mikrokontroler. Sebelum ada perintah pengiriman data dari *PC* terlebih dahulu hasil konversi disimpan pada register yang dimiliki oleh mikrokontroler. Pengiriman karakter berikutnya sebagai pengiriman karakter perintah transfer data dari *PC*. Karakter yang dikirim merupakan alamat tiap *slave*. Bila karakter yang dikirimkan oleh *PC* sama dengan karakter pada mikrokontroler atau pada *slave* yang dituju maka *slave* tersebut akan mengirimkan hasil konversi yang telah tersimpan pada register mikrokontroler menuju *PC*. Hasil konversi akan diolah oleh *PC* menjadi nilai suhu atau kelembaban.



Gambar 3.12 Contoh transfer data pada Slave 1



Gambar 3.13 Diagram alir transfer data tiga *slave*

Prosedur gambar 3.13 dijalankan pada *PC* dengan *software Delphi 7*. Mula-mula koneksi *serial port* dari *PC* menuju *converter* dan *slave* diaktifkan kemudian dipilih *baudrate* dan *port* yang akan mengambil data dari tiap *slave*. Pada saat program dijalankan dilakukan pengiriman karakter untuk mengaktifkan *ADC* pada tiap *slave*. Sesudah karakter untuk mengaktifkan *ADC* dilakukan pengiriman karakter kunci sebagai alamat pada seluruh *slave*. Apabila karakter kunci yang dikirim sesuai dengan *slave* penerima maka *slave* tersebut akan mengirimkan data.

Apabila tidak sesuai maka dilanjutkan dengan karakter berikutnya hingga karakter yang diterima oleh *slave* sesuai dengan yang dikirimkan. Sesudah data diambil, maka data direkam dengan menggunakan *MySQL Delphi* dan ditampilkan pada *PC*.

3.4 Pengujian Sistem Peralatan

Pengujian sistem yang dilakukan meliputi pengujian master, pengujian modul *slave*, pengujian rangkaian sensor, dan pengujian sistem secara keseluruhan.

3.4.1 Pengujian Master

Pengujian master dilakukan untuk menguji fungsi rangkaian sebagai pengubah komunikasi serial antara *PC* dengan mikrokontroler apakah dapat bekerja dengan baik. Selain itu pengujian juga untuk mengetahui apakah data pada *slave* dapat diterima dan ditempatkan secara tepat oleh master. Pengujian dilakukan dengan menghubungkan rangkaian *converter* pada sistem mikrokontroler dan pada port serial *PC*. Pada pengujian rangkaian ini, mikrokontroler terlebih dahulu diisi dengan beberapa karakter sebagai data serta satu karakter kunci. Data yang telah diisikan pada mikrokontroler akan dikirimkan pada *PC* bila ada kesesuaian antara karakter kunci yang dikirim oleh *PC* dengan yang diterima oleh mikrokontroler. Tampilnya data pada *PC* dari mikrokontroler menunjukkan bahwa pengubahan komunikasi serial dari *slave* yang menggunakan RS-485 menuju *PC* yang menggunakan RS-232 dapat dilakukan dan berfungsi dengan baik.

3.4.2 Pengujian Modul Slave

Pengujian rangkaian ini bertujuan mengetahui apakah sistem dari modul *slave* yang telah dirancang dapat bekerja dengan baik. Pengujian meliputi beberapa hal. Diawali dengan pengujian *port A* sebagai masukan dan *port B* sebagai keluaran. Sebelum dilakukan pengujian pada mikrokontroler terlebih dahulu dilakukan pengkonfigurasi pada mikrokontroler. Proses konfigurasi disini dilakukan sesuai prosedur perancangan *software* dengan terlebih dahulu mengkonfigurasi *port A* sebagai masukan dan *port B* sebagai keluaran. Masukan yang diberikan pada *port A* berupa nilai tegangan antara 0-5 volt yang nilainya dapat diubah-ubah. Keluaran pada *port B* berupa *LED* yang akan mengalami perubahan menyala dan mati apabila kita berikan nilai masukan yang berubah-ubah. *LED* menyala berarti bernilai 1 atau dalam kondisi logika tinggi dan *LED* mati berarti bernilai 0 atau dalam kondisi logika rendah.

Pengujian awal terhadap port masukan dan keluaran, bila telah selesai dan berhasil kemudian dilanjutkan dengan pengujian tiga *channel ADC*, dengan masukan analog pada *port A* dan keluaran pada *port B* dan *port C*. Pengujian yang dilakukan hampir sama dengan pengujian awal, hanya saja untuk menguji apakah ketiga *channel ADC* pada *port A* dapat berfungsi ditambahkan *LED* pada *port C* untuk mendukung pengujian *ADC* sebesar 10 bit.

Pengujian *port* masukan dan keluaran sesudah berhasil dilanjutkan pengujian komunikasi serial melalui pin RX/TX mikrokontroler dengan *PC*. *Software Delphi* digunakan sebagai media pendukung komunikasi serial antara mikrokontroler dengan *PC*. Terlebih dahulu sebelum dilakukan pengujian komunikasi serial, kita buat *software* penampil data pada *PC* dengan memanfaatkan komponen-komponen pendukung pada *Delphi* yang sesuai dengan komunikasi serial. Pengujian komunikasi dilakukan dengan cara kita isi mikrokontroler beberapa karakter sebagai data dan satu karakter sebagai karakter kunci, untuk mengambil data yang telah kita isikan pada mikrokontroler kita berikan karakter kunci dari *PC* pada mikrokontroler. Data yang telah kita isikan pada mikrokontroler akan dikirimkan pada *PC* bila ada kesamaan antara karakter kunci yang dikirim oleh *PC* dengan yang ada pada mikrokontroler. Data yang dikirim oleh mikrokontroler selanjutnya akan ditampilkan oleh *PC* sebagai tanda bahwa komunikasi serial antara mikrokontroler dengan *PC* dapat dilakukan.

Pengujian komunikasi serial melalui pin RX/TX mikrokontroler yang telah berhasil dilanjutkan dengan pengujian komunikasi serial dengan memanfaatkan RS485. Pengujian yang dilakukan metodenya sama dengan pengujian komunikasi melalui pin RX/TX mikrokontroler dengan PC, hanya saja pada pengujian komunikasi dengan memanfaatkan RS-485, nilai pada port B yaitu port B1 pada pin 22 harus berubah pada saat proses transfer data. Perubahan nilai ini menunjukkan perintah aktivasi RS-485. Pin RB1 yang terhubung pada pin 2 dan 3 pada RS-485 akan bernilai tinggi atau 1 pada saat transfer data.

3.4.3 Pengujian Rangkaian Sensor

Pengujian rangkaian sensor berfungsi untuk mengetahui apakah kedua sensor tersebut dapat mengukur suhu dan kelembaban sesuai dengan yang telah dirancang. Pengujian dilakukan per bagian dari rangkaian yang telah dirancang. Rangkaian penguat diuji dengan cara memberikan tegangan masukan dan keluaran pada multimeter akan menunjukkan apakah hasil penguatan telah sesuai dengan perancangan. Rangkaian osilator *Wien* diuji dengan melihat bentuk gelombang keluaran pada osiloskop. Rangkaian AC/DC Converter diuji dengan memberikan masukan gelombang sinus dan dihasilkan keluaran berupa garis lurus dengan melihat bentuk keluaran pada osiloskop.

Proses pengujian sensor suhu meliputi pengujian sensor sebelum dan sesudah dikuatkan. Pada pengujian sensor suhu beberapa kali dilakukan pengaturan ulang nilai tegangan pada masukan *inverting*, hal ini dilakukan karena terlalu jauhnya selisih nilai sensor dengan nilai sesudah dikuatkan.

Proses pengujian sensor kelembaban dilakukan uji rangkaian jembatan *Wheatstone*. Pada pengujian rangkaian ini dilakukan penggantian nilai resistor pada bagian sensor kelembaban dari nilai $1k\Omega$ hingga $60k\Omega$. Sesudah pengujian per rangkaian dilanjutkan dengan pengujian keluaran sensor menggunakan multimeter. Nilai keluaran multimeter dibandingkan dengan termometer dan higrometer digital.

Pengujian sensor dilakukan dengan cara memberikan perubahan kondisi ruangan pada laboratorium Instrumentasi dengan memanfaatkan AC pada laboratorium. Kondisi AC diubah dari kondisi normal ruangan ke dingin dan dari kondisi dingin ke kondisi

normal ruangan sebelum dilakukan perubahan kondisi dengan memanfaatkan AC ruangan.

Data perubahan tegangan sensor suhu dan kelembaban menjadi nilai perbandingan antara termometer dan higrometer digital. Hasil perbandingan berupa persamaan sebagai persamaan dalam *software master* untuk menampilkan nilai suhu dan kelembaban pada *PC*. Kemudian rangkaian sensor dihubungkan dengan mikrokontroler dan *PC* sebagai penampil nilai keluaran sensor dengan keluaran berupa suhu dan kelembaban.

3.4.4 Pengujian Sistem Keseluruhan

Pengujian dilakukan untuk mengetahui apakah kerja dari sistem secara keseluruhan dapat bekerja sesuai dengan perancangan. Pengujian sistem pengukuran dan monitoring yang terdiri dari *master* dan *slave* dilakukan secara bersamaan. Pada *slave* yang terdiri dari sensor dan mikrokontroler dihubungkan dengan RS-232 To RS-485 *converter* yang telah terhubung dengan *port* serial dari *PC*. Pengujian dilakukan dengan cara membuka koneksi awal antara master dengan *slave* melalui *master*. Bila koneksi berhasil maka suhu dan kelembaban yang terukur akan ditampilkan oleh *PC*.

Pengambilan data dilakukan untuk mengamati hasil perancangan secara keseluruhan. Apabila sistem telah dapat bekerja, maka data yang telah dikirimkan oleh tiap *slave* menuju master akan ditampilkan pada *PC* berupa nilai suhu dan kelembaban. Nilai suhu dan kelembaban dari tiap *slave* yang berbeda tersebut akan ditampilkan sesuai dengan urutan *slave*. Selain itu data hasil monitoring yang telah diambil dapat diakses ulang oleh *master* yang berfungsi sebagai media penyimpanan data hasil monitoring.

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Alat

Pengujian alat dimaksudkan untuk mengetahui sampai sejauh mana sistem kerja dari sensor hingga pendukung komunikasi multidrop RS-485 dapat bekerja.

Pengujian disini meliputi pengujian ketiga sensor itu sendiri, pengujian pengkondisi sinyal, pengujian data akuisisi oleh mikrokontroler dan pengujian alat secara keseluruhan sebagai pengambilan data.

a. Pengujian Sensor Suhu

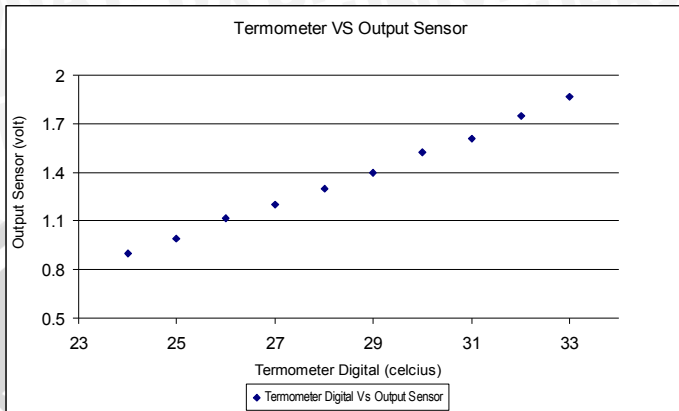
Berdasarkan beberapa pengujian yang telah dilakukan, sensor suhu dengan pengondisi sinyal dapat berfungsi sesuai dengan yang diharapkan meskipun pada data didapat nilai yang tidak stabil. Pada pengaturan nilai tegangan, kita pergunakan persamaan matematis pembagi tegangan, yaitu

$$v_{out} = \frac{R_2}{R_2 + R_1} \times v_{input} \quad (4.1)$$

dengan R_2 (trimpot) sebesar $1k\Omega$ dan R_1 sebesar $10k\Omega$ serta tegangan input 5V. Sesudah dilakukan pengurangan maka nilai keluaran dari sensor setelah melalui proses penguatan menjadi persamaan 4.2.

$$v_{out} = (v_{sensor} - 150\text{mV}) \times 10 \quad (4.2)$$

Bila tiap kenaikan suhu menghasilkan tegangan sebesar 10mV, maka pengkondisi sinyal dikondisikan pada saat suhu 15°C memiliki nilai 0 Volt.



Gambar 4.1 Grafik hasil pengujian sensor suhu dengan keluaran berupa tegangan yang dibandingkan dengan termometer digital.

Pada saat dilakukan pengujian terdapat kondisi dimana respon sensor suhu lebih lambat dibandingkan dengan termometer digital.

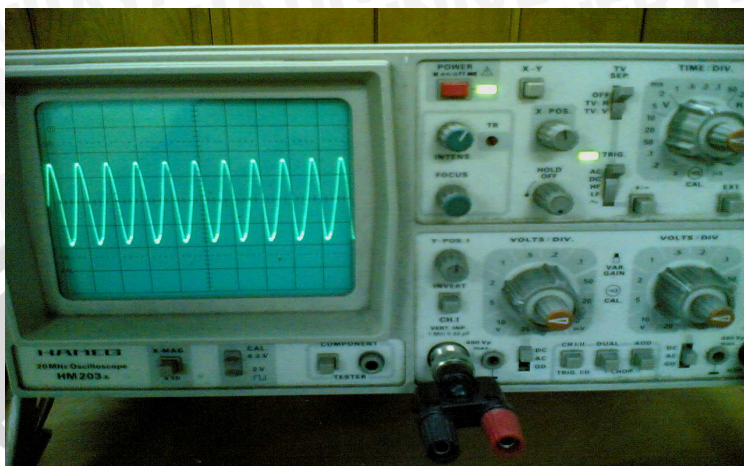
b. Pengujian Sensor Kelembaban

Berdasarkan hasil pengujian diperoleh nilai-nilai yang hampir sama antara pengujian rangkaian secara langsung dengan pengujian secara matematis menggunakan persamaan 4.3.

$$V_{out} = \left[\left(\frac{R_3}{R_1 + R_3} \right) - \left(\frac{R_4}{R_2 + R_4} \right) \right] \times V_{in} \quad (4.3)$$

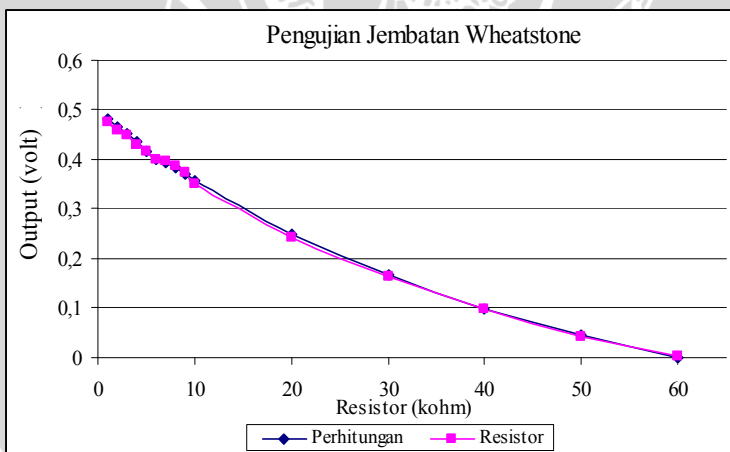
Dimana nilai R_4 merupakan nilai yang diubah mulai dari $1k\Omega$ sampai dengan $60k\Omega$ dan nilai tegangan input $1VAC$.

Nilai $1VAC$ dengan frekuensi $1kHz$ diperoleh dari rangkaian Osilator *Wien*. Sebelum dilakukan pengujian pada rangkaian jembatan *Wheatstone* terlebih dahulu dilakukan pengujian rangkaian osilator. Berdasarkan pengujian rangkaian osilator sebelum memasuki rangkaian pembagi tegangan diperoleh hasil sebagaimana pada gambar 4.2.



Gambar 4.2 Hasil keluaran rangkaian osilator *Wien* berfrekuensi 1kHz sebelum memasuki rangkaian pembagi tegangan.

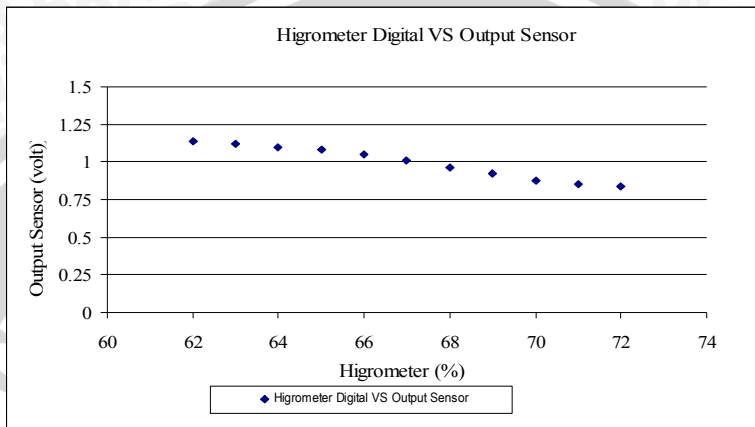
Grafik pada gambar 4.3 menggambarkan adanya kesesuaian meskipun tidak ideal antara pengujian menggunakan resistor dengan nilai yang diperoleh secara perhitungan.



Gambar 4.3 Grafik hasil pengujian jembatan *Wheatstone*

Pada pengujian sensor kelembaban, selama proses pengujian nilai perubahan yang ditampilkan pada multimeter dibandingkan dengan

higrometer digital. Berdasarkan pengujian, terdapat kondisi dimana nilai yang ditampilkan pada multimeter mengalami kenaikan maupun penurunan secara berubah-ubah.



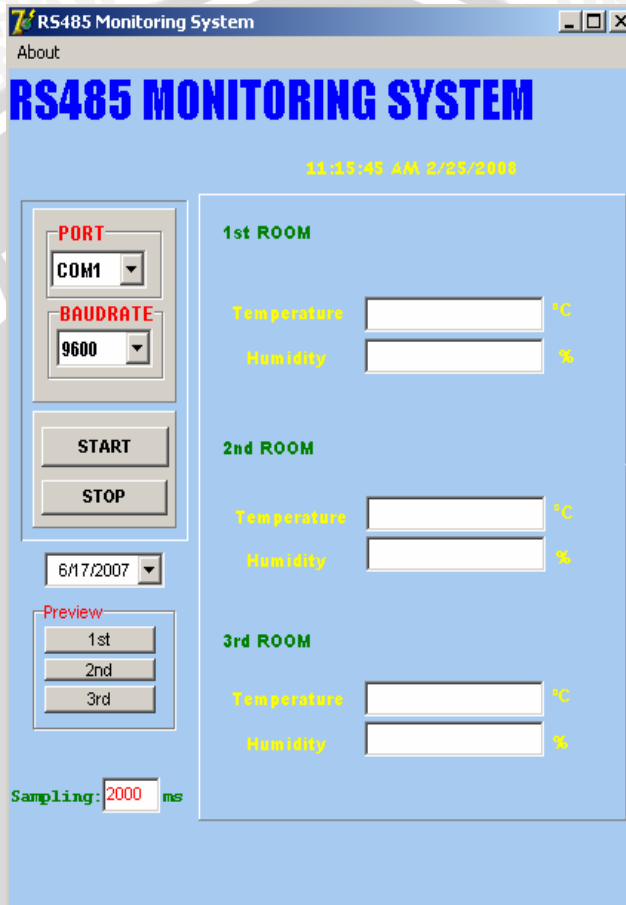
Gambar 4.4 Grafik hasil pengujian sensor kelembaban dengan keluaran berupa tegangan yang dibandingkan dengan higrometer digital.

4.2 Pengujian *Software*

Pengujian *software* meliputi pengujian komunikasi antara PC dengan seluruh modul *slave*. Dari pengujian *software* yang telah dilakukan, komunikasi antara tiap *slave* dengan PC dapat dilakukan. Hal ini terbukti dari tampilnya nilai dari tiap *slave* pada PC. Disamping itu PC dapat merekam nilai dari tiap *slave*.

Proses kerja dari *software* yang telah dibuat mula-mula diatur pendukung komunikasi antara *master* dengan *slave*. Yaitu, koneksi *serial port* pada PC yang digunakan dan *baudrate* yang sesuai antara PC dengan mikrokontroler dalam berkomunikasi. Kemudian mula-mula dari PC mengirimkan karakter kunci yang telah ditentukan untuk mengaktifkan *ADC* yang terdapat pada tiap *slave* secara serentak. *ADC* yang telah diaktifkan akan mulai bekerja dan menyimpan hasil konversi pada memori internal yang terdapat pada mikrokontroler. Kemudian dikirimkan karakter kunci yang identik untuk meminta data dari tiap *slave*, apabila karakter kunci yang diterima oleh *slave* sama dengan yang telah dikirimkan oleh PC, maka data dari *slave* yang sebelumnya disimpan pada mikrokontroler

akan ditransfer untuk disimpan pada *PC*. Data yang telah dikirim pada *PC* akan diolah dan ditampilkan berdasarkan posisi *slave*.



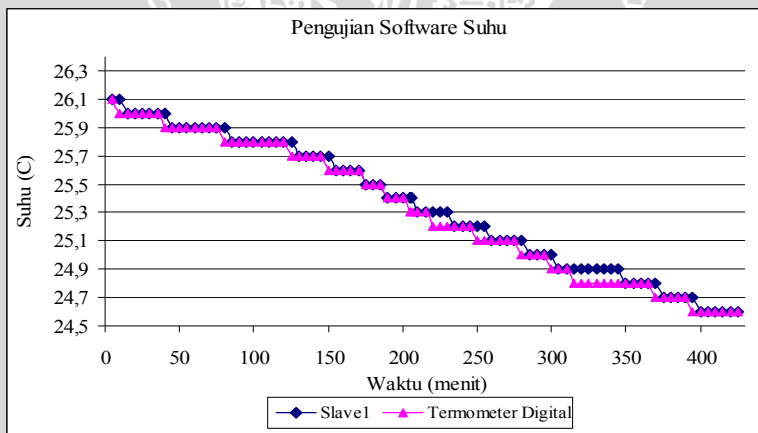
Gambar 4.5 Tampilan software

Tampilan gambar 4.5 menunjukkan perbedaan penempatan data yang diterima oleh masing-masing *slave*. Angka 1st, 2nd dan 3rd menunjukkan data yang ditampilkan dari *slave* 1, 2 dan 3. Pemilihan *serial port* pada *PC* untuk membuat komunikasi serial yang dilakukan antara *master* dengan *slave* lebih fleksibel sehingga tidak bergantung pada satu *COM*. Adanya pengaturan *baudrate* pada tampilan merupakan gambaran diperlukannya kesesuaian *baudrate*

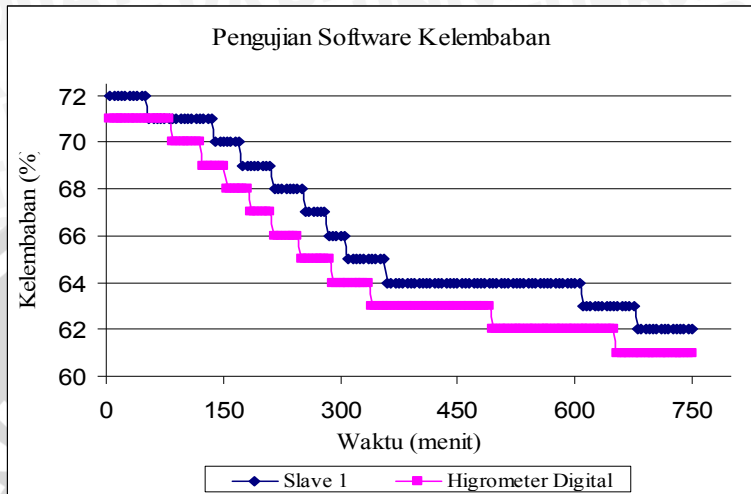
yang harus dimiliki oleh mikrokontroler dengan PC dalam berkomunikasi. Berdasarkan beberapa kali pengujian *baudrate* yang dibutuhkan antara keduanya dan berdasarkan literatur, *baudrate* sebesar 9600bps telah mampu mendukung komunikasi antara *master* dengan *slave* tanpa adanya keterlambatan pengiriman data dari *slave* menuju *master*.

Karena *master* berfungsi sebagai media penyimpan data hasil monitoring, maka adanya kolom tanggal/bulan/tahun dengan tombol 1st, 2nd dan 3rd berfungsi sebagai penampil data hasil monitoring yang telah dilakukan oleh tiap *slave*.

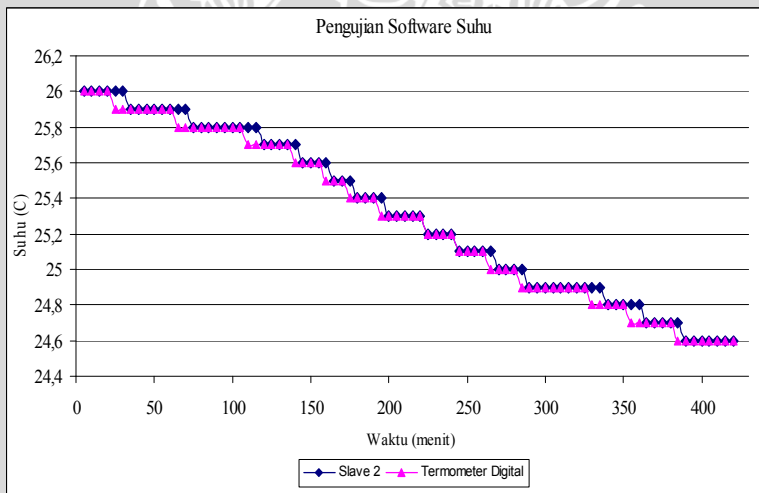
PC akan mengatur kerja dari RS-485 melalui pin *RTS Control* dari *port* serial. Dimana untuk melakukan pengiriman data maka *RTS* diset berlogika rendah atau bernilai 0 sehingga RS-485 memiliki logika tinggi, sedangkan saat pengiriman selesai maka *RTS* diset berlogika tinggi atau bernilai 1 sehingga RS-485 berlogika rendah. Hal ini dilakukan pula oleh *slave* melalui mikrokontroler, pada saat *slave* melakukan pengiriman data maka mikrokontroler akan mengaktifkan pin *DE* pada RS-485 dengan memberikan logika tinggi pada RS-485 melalui port B. Bila pengiriman telah selesai maka mikrokontroler akan memberikan nilai logika rendah pada RS-485.



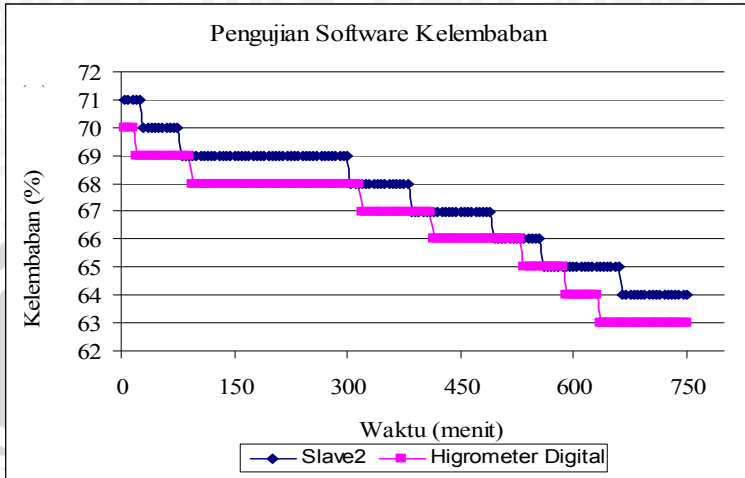
Gambar 4.6 Grafik hasil pengujian *software* berupa nilai suhu yang dikirim oleh *slave* 1 dibandingkan terhadap termometer digital.



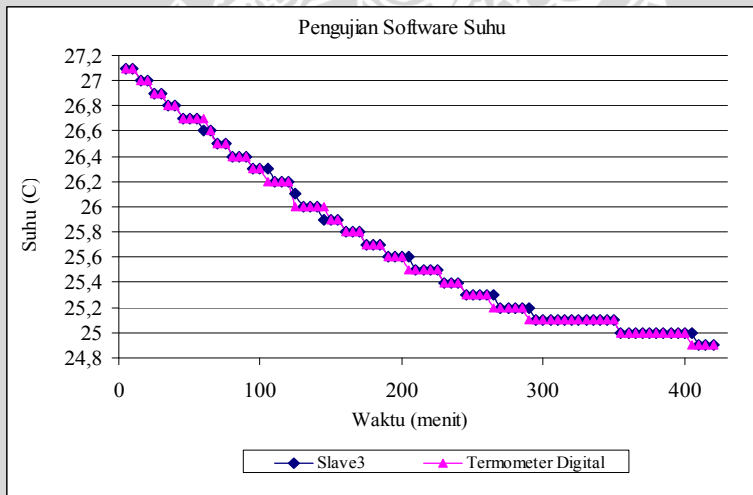
Gambar 4.7 Grafik hasil pengujian *software* berupa nilai kelembaban yang dikirim oleh *slave 1* dibandingkan dengan higrometer digital.



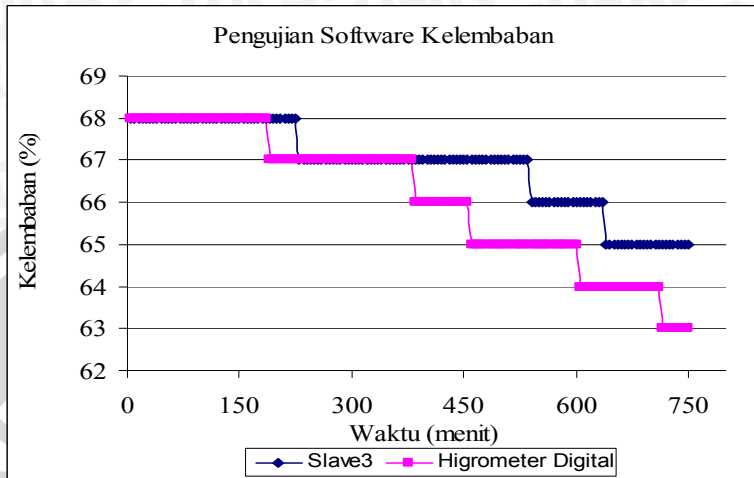
Gambar 4.8 Grafik hasil pengujian *software* berupa nilai suhu yang dikirim oleh *slave 2* dibandingkan dengan termometer digital.



Gambar 4.9 Grafik hasil pengujian *software* berupa nilai kelembaban yang dikirim oleh *slave 2* dibandingkan dengan higrometer digital.



Gambar 4.10 Grafik hasil pengujian *software* berupa nilai suhu yang dikirim oleh *slave 3* dibandingkan dengan termometer digital.



Gambar 4.11 Grafik hasil pengujian *software* berupa nilai kelembaban yang dikirim oleh *slave 3* dibandingkan dengan higrometer digital.

Hasil pengujian *software* suhu yang ditampilkan oleh *PC* berupa grafik pada gambar 4.6, gambar 4.8 dan gambar 4.10. Hasil pengujian menunjukkan adanya kesesuaian antara nilai suhu dari ketiga *slave* dengan termometer digital. Kesesuaian ditunjukkan oleh grafik berupa titik-titik *slave* dan termometer digital yang saling berhimpit. Gambar 4.6, gambar 4.8, dan gambar 4.10 menunjukkan keterlambatan respon dari tiap *slave* dibandingkan dengan termometer digital meskipun setelah beberapa saat memiliki kesesuaian.

Hasil pengujian *software* kelembaban yang ditampilkan oleh *PC* berupa grafik pada gambar 4.7, gambar 4.9, dan 4.11. Hasil pengujian menunjukkan perubahan nilai kelembaban yang dikirimkan oleh ketiga *slave* memiliki kesesuaian dengan higrometer digital meskipun nilai yang ditampilkan menggambarkan respon yang lebih lambat. Kesesuaian ditunjukkan dengan titik-titik *slave* dan higrometer digital yang saling berhimpit.

Selisih waktu yang dimiliki oleh *slave* hasil perancangan dengan termometer dan higrometer digital diakibatkan oleh waktu yang dibutuhkan oleh sensor untuk mencapai kondisi ideal pada saat proses pengukuran kondisi lingkungan. *Datasheet* sensor suhu LM35 memberikan informasi bahwa sensor LM35 membutuhkan waktu

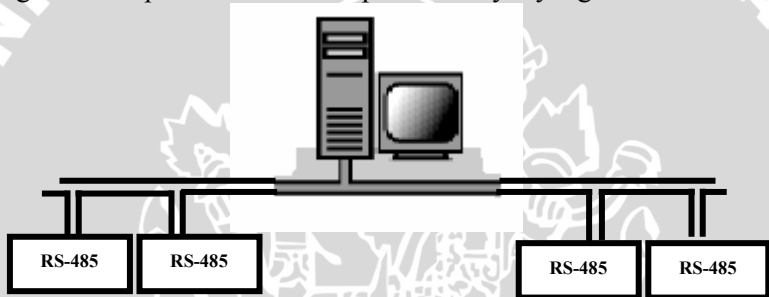
selama tiga menit untuk mencapai kondisi ideal sesuai dengan kondisi lingkungan pada proses pengukuran suhu. Respon waktu sebesar tiga menit menyebabkan adanya keterlambatan perubahan antara hasil pengukuran yang ditampilkan dari tiap *slave* dengan termometer digital. *Datasheet* dari sensor kelembaban HS15P memberikan informasi bahwa sensor HS15P membutuhkan waktu minimal sepuluh menit untuk mencapai kondisi ideal dengan lingkungan pada saat pengukuran kelembaban lingkungan. Waktu respon sebesar sepuluh menit mengakibatkan data yang dikirim oleh *slave* jauh lebih lambat daripada higrometer digital.

RS-232 (Recommended Standard-232) merupakan standar komunikasi yang dikeluarkan *Electronic Industries Association* (EIA) untuk menghubungkan perangkat serial secara *point to point*. Standar komunikasi serial memiliki jangkauan yang jauh dan kecepatan komunikasi data relatif rendah. Sinyal RS-232 dikirimkan dalam bentuk level tegangan yang bergantung pada *ground* dan bekerja dengan baik untuk komunikasi *point to point* atau satu *port* hanya dapat berkomunikasi dengan satu perangkat seperti COM1 hanya untuk *mouse* dan COM2 untuk modem. RS-232 membutuhkan sebuah *ground* bersama antara *PC* dengan perangkat yang terhubung. Panjang kabel terbatas 100-200 feet untuk komunikasi asinkron dan 50 feet untuk komunikasi sinkron (www.hw-server.com/docs/rs485.html).

Standar RS-232 memiliki beberapa keterbatasan seperti adanya *ground* bersama antara *Data Terminal Equipment* (DTE) dan *Data Circuit-Terminating Equipment* (DCE) sehingga sinyal pada jalur yang sama cukup mustahil untuk terhindar dari noise. Selain itu mengakibatkan keterbatasan panjang kabel yang kurang lebih terbatas hanya 30 sampai 60 meter. Level tegangan yang bergantung pada *ground* hanya sesuai untuk komunikasi *point to point* atau 1 *transmitter* untuk 1 *receiver*.

RS-485 dikembangkan oleh EIA (*Electronics Industry Association*) untuk kebutuhan komunikasi serial yang dapat mencapai jarak lebih jauh dengan jumlah titik percabangan yang lebih banyak. RS-485 sering digunakan dalam sistem komunikasi banyak titik atau dikenal dengan istilah *multidrop* atau *multi point* RS-485. RS-485 mendukung atau sesuai untuk 32 *driver* dan 32 *receiver*. RS-485 merupakan koneksi serial *multipoint*, biasanya memanfaatkan dua kabel dan komunikasinya bersifat *half-duplex*

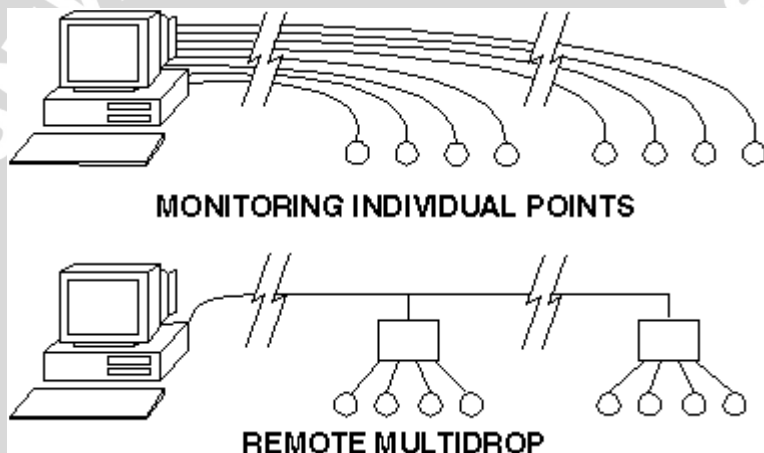
atau proses transfer data harus dilakukan secara bergantian dan tidak dapat dilakukan dalam satu waktu. Standar yang dimiliki berupa selisih dalam pengiriman sinyal, selisih antara tegangan kabel atau jalur adalah apa yang disebut data. Satu polaritas tegangan mengindikasikan logika 1 dan polaritas lawan sebagai logika 0. Selisih minimum yang harus dimiliki sebesar 0,2 volt untuk kinerja yang bagus. RS-485 memiliki satu *driver* dan satu *transceiver*, kondisi ini berbeda dengan RS-422 yang memiliki satu driver yang tidak dapat dimatikan. Pada RS-485 *driver* harus dikondisikan dalam mode kirim dengan memberikan sinyal pada *driver*. Jalur komunikasi umumnya menggunakan topologi linear dengan dua kabel. Untuk jumlah yang lebih banyak atau jarak yang lebih jauh bisa dipergunakan *repeater* untuk memperkuat sinyal yang dihasilkan.



Gambar 4.12 Topologi linear dua kabel untuk RS-485

Converter digunakan untuk mengubah karakteristik kelistrikan satu bentuk standar komunikasi menjadi bentuk yang lain dengan mempertimbangkan standar komunikasi yang dipilih. Misalkan dengan RS-232 To RS-485 *Converter* bisa dihubungkan PC yang menggunakan RS-232 bersifat *full duplex* menjadi RS-485 bersifat *half duplex* dengan banyak titik sejauh 4000 *feet* atau 1,2 km. *Converter* akan melewati data melalui antarmuka yang dipergunakan tanpa merubah waktu dan atau protokol. Dengan memanfaatkan *software*, RS-485 *transmitter* akan aktif ketika ada data yang terdeteksi pada port RS-232 dan menonaktifkan kembali mode penerima setelah sebuah karakter dikirimkan. Ketika *fullduplex* dirubah menjadi *half duplex* hanya satu perangkat yang mengirim data pada satu waktu. Dengan cara ini masalah pewaktuan dapat dihindari dan kecepatan komunikasi dapat dicapai.

Multidrop RS-485 merupakan sistem jaringan banyak titik yang memanfaatkan kerja dari RS-485. Biasa dikenal pula dengan istilah *single master multi slave* atau satu pusat dengan banyak percabangan. Dengan metode ini, bisa dilakukan pengamatan pada sejumlah objek melalui satu pusat. Pengamatan yang dilakukan dengan *multidrop* hanya membutuhkan satu jalur utama dari *master* menuju *slave*. Hal ini berbeda dengan sistem *point to point* yang membutuhkan sejumlah jalur dari pusat pengamatan menuju percabangan. Dengan sistem *multidrop*, tingkat efisiensi dalam pengamatan dapat dicapai lebih maksimal.



Gambar 4.13 Multipoint dan individual point

Sistem multidrop membutuhkan protokol untuk mengatur pengalaman. Protokol akan mengatur komunikasi antara *master* dengan *slave*. Setiap *slave* yang terhubung dengan *master* memiliki alamat tersendiri yang unik. Dengan alamat yang unik adanya kesalahan pengambilan data dari *slave* yang berbeda dapat dihindari. Umumnya RS-485 memanfaatkan protokol asinkron (UART) untuk mendukung komunikasi antara *master* dengan *slave*. Protokol asinkron memiliki kelebihan pada perancangan *software* yang tidak memerlukan pengaturan sinkronisasi *clock* pada kedua sisi pengiriman data.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini telah dihasilkan peralatan yang mendukung komunikasi serial dengan sistem terdistribusi multidrop RS485. Dalam sistem terdistribusi multidrop RS-485, komponen pendukung komunikasi dibagi menjadi dua bagian, yaitu *PC* sebagai *master* yang berfungsi sebagai media pengontrol fungsi *datalogger* keseluruhan sistem dan mikrokontroler sebagai *slave* yang berfungsi sebagai media pengambil data pada objek yang dimonitoring.

Penggunaan karakter kunci dapat digunakan sebagai penanda dan pengatur penerimaan data dari tiap *slave*. Dengan adanya karakter kunci tersebut, data dari tiap *channel* yang dikirim dari tiap *slave* dapat ditempatkan dengan tepat pada *PC*.

5.2 Saran

Perlu dilakukan penyempurnaan *software* lebih lanjut yang dapat mendukung segala bentuk kebutuhan aplikasi. Penyempurnaan dilakukan tidak hanya pada sistem komunikasi tetapi juga pada kesesuaian antara hasil perancangan dengan sarana kalibrasi yang dipakai pada saat proses kalibrasi.

Perlu dilakukan penggantian sensor suhu dan kelembaban yang lebih baik. Sensor LM35 dan sensor HS15P perlu digantikan dengan sensor yang memiliki waktu respon lebih cepat. Kedua sensor dapat digantikan dengan sensor yang dapat menghasilkan nilai digital tanpa menggunakan ADC dan telah terkalibrasi.

DAFTAR PUSTAKA

- Axelson, J. 1999. **Serial Port Complete: Programming and Circuits For RS-232 and RS-485 Links and Networks.** Lakeview Research. Madison
- Budiharto, Widodo Ssi, Mkom. 2004. **Interfacing Komputer dan Mikrokontroler.** PT. Elex Media Komputindo. Jakarta
- GE Sensing. 2005. **Types HS12P and HS15P Specifications.** General Electric Company. USA
- Giancoli, Douglas.C. 2001. **Fisika.** Erlangga. Jakarta
- Lyons, Jeff. 1999. **Distributed Monitoring And Control Modelling For A Laboratory Three Tank System.** Electrical Engineering Vanderbilt University. Tennessee
- Malik, Moh.Ibnu. 2003. **Belajar Mikrokontroler PIC 16F84.** Gava Media. Yogyakarta
- National Semiconductor. 1995. **LM35 Precision Centrigade Temperature Sensor.** National Semiconductor Corporation. USA
- Santoso, D. R. 2005. **Development of Piezoelectric Sensor Module And System For SHM Instrumentation.** Indonesia Scientific Meeting. Nagoya-Japan
- Tipler, Paul. A. 1998. **Fisika Untuk Sains Dan Teknik.** Erlangga. Jakarta
- <http://www.hw-server.com/docs/rs485.html>,
tanggal akses : 14 April 2007

Lampiran1
Data Hasil Pengujian Rangkaian

1.1 Data Uji Jembatan *Wheatstone*

Resistor(k Ω)	Voutput (Vpp)	Perhitungan(Vpp)
1	0,475	0,483
2	0,460	0,467
3	0,449	0,452
4	0,430	0,437
5	0,418	0,416
6	0,401	0,400
7	0,396	0,395
8	0,388	0,382
9	0,375	0,369
10	0,352	0,357
20	0,244	0,250
30	0,163	0,166
40	0,098	0,100
50	0,043	0,045
60	0,003	0,000

1.2 Data Uji Rangkaian Penguat Sensor Suhu LM35

Vin (volt)	Vout (Volt)	Perhitungan
0.15	-0.03	0
0.2	0.42	0.5
0.25	0.92	1
0.3	1.43	1.5
0.35	1.93	2
0.4	2.38	2.5
0.45	2.9	3
0.5	3.35	3.5

1.3 Data Uji Sensor Suhu

Termometer (celcius)	Output sensor (volt)
24	0.9
25	0.99
26	1.12
27	1.2
28	1.3
29	1.4
30	1.52
31	1.61
32	1.75
33	1.87

1.4 Data Uji Sensor Kelembaban

Higrometer (%)	Output sensor (Volt)
62	1.14
63	1.12
64	1.1
65	1.08
66	1.05
67	1.01
68	0.96
69	0.92
70	0.88
71	0.85
72	0.84

Lampiran2

2.1 *Source Code Master* (Delphi7)

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,

Dialogs, StdCtrls, CPort, CPortCtl, Gauges, ExtCtrls, XPMAN, Grids,
Menus,

ComCtrls, jpeg, Buttons, DB, DBGrids, mySQLDbTables, TeEngine,
Series,

TeeProcs, Chart, DbChart, RpCon, RpConDS, RpDefine, shellapi, RpRave;

type

TForm1 = class(TForm)

StringGrid1: TStringGrid;

StringGrid3: TStringGrid;

Timer1: TTimer;

Label6: TLabel;

StatusBar1: TStatusBar;

Bevel2: TBevel;

Label1: TLabel;

Panel1: TPanel;

Panel4: TPanel;

Label7: TLabel;

Timer2: TTimer;

Timer3: TTimer;

Label9: TLabel;

Button1: TButton;

Button4: TButton;

Button3: TButton;

Panel3: TPanel;

DBGrid1: TDBGrid;

DBGrid2: TDBGrid;

DBGrid3: TDBGrid;

db1: TMySQLDatabase;

qdata1: TMySQLQuery;

qdata1no: TWordField;

qdata1waktu: TDateTimeField;

qdata1suhu1: TFloatField;

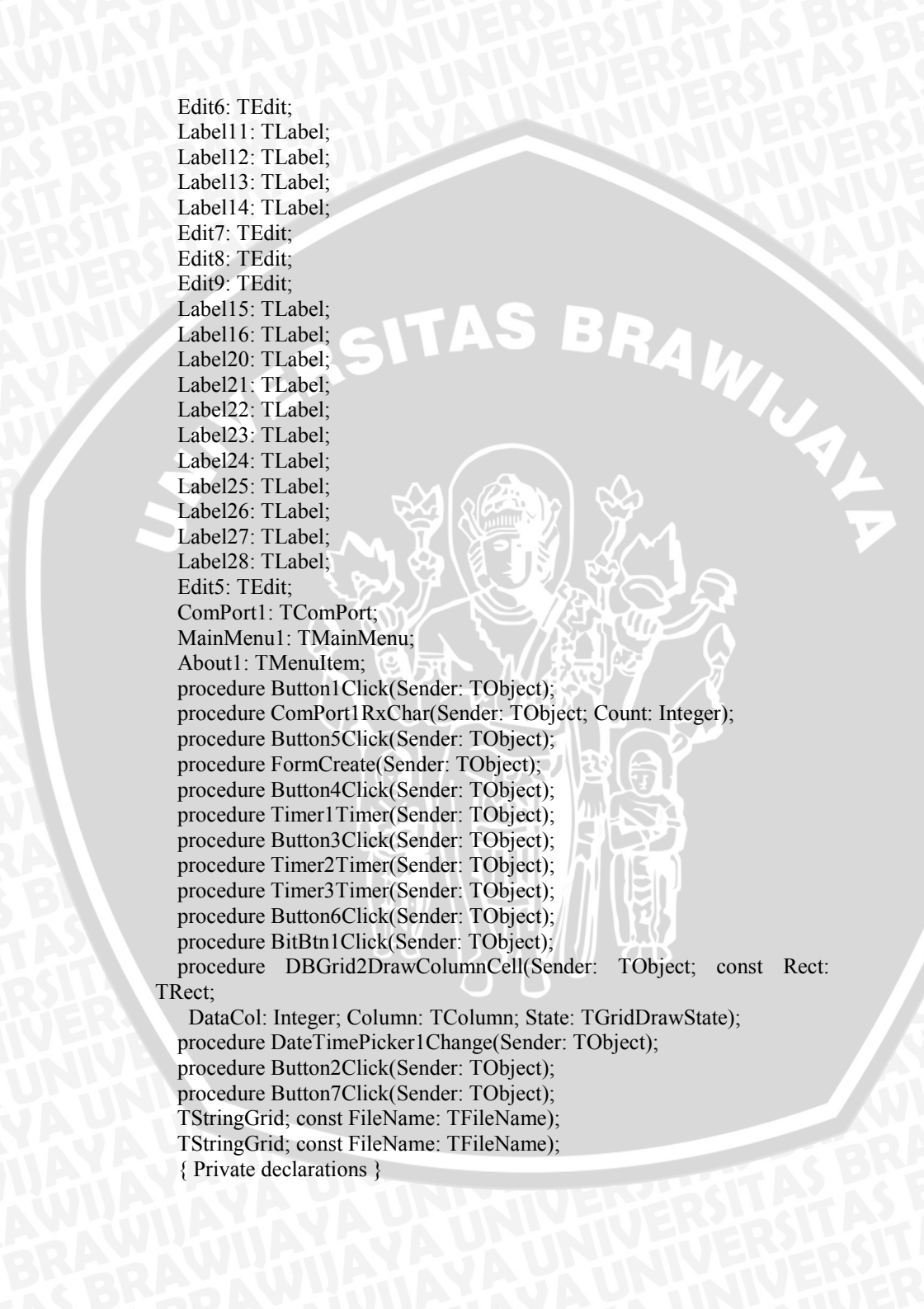
qdata1kelembaban1: TFloatField;

qdata1tekanan1: TFloatField;

DataSource1: TDataSource;

DataSource2: TDataSource;

qdata2: TmySQLQuery;
qdata2no: TWordField;
qdata2waktu: TDateTimeField;
qdata2suhu2: TFloatField;
qdata2kelembaban2: TFloatField;
qdata2tekanan2: TFloatField;
qdata3: TmySQLQuery;
DataSource3: TDataSource;
qdata3no: TWordField;
qdata3waktu: TDateTimeField;
qdata3suhu3: TFloatField;
qdata3kelembaban3: TFloatField;
qdata3tekanan3: TFloatField;
q1: TmySQLQuery;
q2: TmySQLQuery;
Label3: TLabel;
StringGrid2: TStringGrid;
Label4: TLabel;
Label5: TLabel;
Label8: TLabel;
q3: TmySQLQuery;
Edit1: TEdit;
PORT: TGroupBox;
ComComboBox1: TComComboBox;
GroupBox1: TGroupBox;
ComComboBox2: TComComboBox;
DateTimePicker1: TDateTimePicker;
RvProject1: TRvProject;
RvDataSetConnection1: TRvDataSetConnection;
GroupBox2: TGroupBox;
Button2: TButton;
RvProject2: TRvProject;
RvDataSetConnection2: TRvDataSetConnection;
Button5: TButton;
RvProject3: TRvProject;
RvDataSetConnection3: TRvDataSetConnection;
Button6: TButton;
Button7: TButton;
Edit2: TEdit;
Edit3: TEdit;
Label2: TLabel;
Label10: TLabel;
Edit4: TEdit;



```

Edit6: TEdit;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Label15: TLabel;
Label16: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Edit5: TEdit;
ComPort1: TComPort;
MainMenu: TMainMenu;
About1: TMenuItem;
procedure Button1Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure Button5Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Timer3Timer(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure DBGrid2DrawColumnCell(Sender: TObject; const Rect:
TRect;
  DataCol: Integer; Column: TColumn; State: TGridDrawState);
procedure DateTimePicker1Change(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
TStringGrid; const FileName: TFileName);
TStringGrid; const FileName: TFileName);
{ Private declarations }

```

```

public
{ Public declarations }
procedure tambah(suhu1,kel1,tek1:double;detect:integer);
procedure tambah2(suhu2,kel2,tek2:double;detect:integer);
procedure tambah3(suhu3,kel3,tek3:double;detect:integer);
end;

var
Form1: TForm1;
bar,sof,sof2,bare,line1,line2,line3,line4,line5,line6,line7,line8,
line9,read,line,kol,bir,adr,i,k,no,j,l,m,low,ganjil,high,berat,iTmp:integer;
db,g:variant;
data,eep,f:TextFile;
nilai,digital,lembab,suhu:real;
strTemp: String;
del: longint;
line1_2:integer;
implementation
{$R *.dfm}
uses comobj;

procedure hitungnilai(high:integer;low:integer);
begin
    nilai:=(high*256)+low;    //digital
    digital:=(nilai/1023)*5;  //tegangan
    lembab:=(digital*-29)+96.475;
    suhu:=(digital*9.5161)+15.596;
end;
{procedure komunikasi RS-485}
procedure TForm1.Button1Click(Sender: TObject);
var
    kirim:char;
begin
    timer2.Interval:=strtoint(edit9.Text);
    timer2.Enabled:=true;
    comport1.Connected:=true;

    comport1.SetRTS(false); //output enable
    comport1.WriteStr(edit5.Text); //key character
    button1.Caption:='DISCONNECT';
    button4.Enabled:=false;
    comport1.SetRTS(true); //input enable

    button1.Caption:='CONNECT';

```

```

button4.Enabled:=true;
comport1.Connected:=true;
button4.Enabled:=true;
comcombobox1.Enabled:=true;
comcombobox2.Enabled:=true;
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
hasil,scanner:byte;
k,b:integer;
suhu1,lembab1,digital1,suhu2,lembab2,suhu3,lembab3:string;
begin
comport1.SetRTS(true);
comport1.Read(hasil,count);
stringgrid3.Cells[kol,1]:=inttostr(hasil);
inc(kol);

If kol=7 then
with stringgrid3 do
begin
// suhu
hitungnilai(strtoint(Cells[kol-2,bar]),strtoint(Cells[kol-1,bar]));
stringgrid2.Cells[sof,line1]:=floattostrf(suhu,ffFixed, 3, 3);;
edit2.Text:=stringgrid2.Cells[1,line1];
edit4.Text:=stringgrid2.Cells[4,line1];
edit7.Text:=stringgrid2.Cells[7,line1];
// lembab
inc(sof);
hitungnilai(strtoint(Cells[kol-4,bar]),strtoint(Cells[kol-3,bar]));
stringgrid2.Cells[sof,line1]:=floattostrf(lembab,ffFixed, 0, 0);
edit3.Text:=stringgrid2.Cells[2,line1];
edit6.Text:=stringgrid2.Cells[5,line1];
edit8.Text:=stringgrid2.Cells[8,line1];
// sensor lain
inc(sof);
hitungnilai(strtoint(Cells[kol-6,bar]),strtoint(Cells[kol-5,bar]));
suhu1:=floattostrf(digital,ffFixed, 2, 2);
stringgrid2.Cells[sof,line1]:=suhu1;

inc(sof);
kol:=1;

```

```

if edit5.Text='M' then //slave 1
begin
edit5.Text:='N'; //slave2
button1.Click;
end
else if edit5.Text='N' then
begin
edit5.Text:='O'; //slave 3
button1.Click;
end
//
end
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
qdata2.Close; //for Mysql Database
qdata2.SQL.Clear;
qdata2.SQL.Add('select * from monitoring where no>-1 and detect =2 and
date(waktu)=date(:waktu2)');
qdata2.ParamByName('waktu2').AsDate:=datetimepicker1.Date;
qdata2.ExecSQL;
qdata2.Open;
rvproject2.projectfile:='prin2.rav'; //nama file simpan data di MySQL
rvproject2.Execute;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
timer2.Enabled:=false;
comport1.Connected:=false;

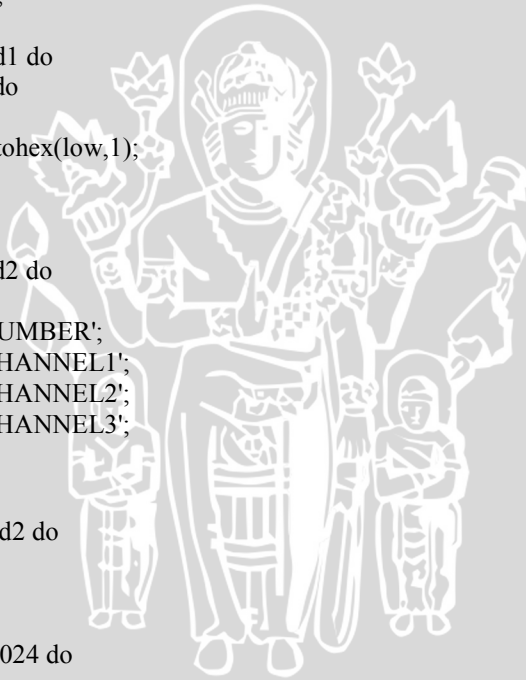
bir:=0;
line:=1;
line1:=1;
line2:=1;
line3:=1;
line4:=1;
line5:=1;
line6:=1;
line7:=1;
line8:=1;
line9:=1;
line1_2:=1;

```

```

sof:=1;
m:=1;
bar:=1;
kol:=1;
k:=1;
adr:=$0;
low:=0;
begin
  with StringGrid1 do
    for i:=1 to 2033 do
      begin
        cells [0,0]:='ADDRESS';
        cells[0,i]:=inttohex(adr,4);
        adr:=adr+$10;
      end;
      with Stringgrid1 do
        for k:=1 to 16 do
          begin
            cells[k,0]:=inttohex(low,1);
            low:=low+1;
          end;
        end;
        with StringGrid2 do
          begin
            Cells[0,0]:='NUMBER';
            Cells[1,0]:='CHANNEL1';
            Cells[2,0]:='CHANNEL2';
            Cells[3,0]:='CHANNEL3';
          end;
        end;
      begin
        with StringGrid2 do
          begin
            no:=1;
            begin
              for j:=1 to 1024 do
                begin
                  Cells[0,j]:=' '+inttostr(no);
                  no:=no+1;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```



procedure TForm1.Button4Click(Sender: TObject);

begin

button1.Enabled:=true;

db1.CleanupInstance;

comcombobox1.Enabled:=true;

comcombobox2.Enabled:=true;

for bar:=1 to 129 do

begin

for kol:=1 to 16 do

begin

Stringgrid1.Cells[kol,bar]:=' ';

Stringgrid3.Cells[kol,bar]:=' ';

end;

end;

for line:=1 to 1024 do

begin

Stringgrid2.Cells[1,line]:=' ';

Stringgrid2.Cells[2,line]:=' ';

Stringgrid2.Cells[3,line]:=' ';

end;

kol:=1;

bar:=1;

line:=1;

line1:=1;

line2:=1;

line3:=1;

line4:=1;

line5:=1;

line6:=1;

line7:=1;

line8:=1;

line9:=1;

m:=1;

comport1.Connected:=false;

button1.Caption:='CONNECT';

bir:=0;

statusbar1.panels[0].Text:='Disconnect';

end;



```
procedure TForm1.Timer1Timer(Sender: TObject);  
{untuk mengatur waktu sampling, berapa detik sekali}  
begin  
  label6.Caption:=timetostr(time);  
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
  timer2.Enabled:=false;  
  button1.Caption:='CONNECT';  
  button4.Enabled:=true;  
end;
```

```
procedure TForm1.Timer2Timer(Sender: TObject);  
begin
```

```
  tambah(strtfloat(stringgrid2.Cells[1,line1_2]),  
    strtfloat(stringgrid2.Cells[2,line1_2]),  
    strtfloat(stringgrid2.Cells[3,line1_2],1);  
  tambah2(strtfloat(stringgrid2.Cells[4,line1_2]),  
    strtfloat(stringgrid2.Cells[5,line1_2]),  
    strtfloat(stringgrid2.Cells[6,line1_2],2);  
  tambah3(strtfloat(stringgrid2.Cells[7,line1_2]),  
    strtfloat(stringgrid2.Cells[8,line1_2]),  
    strtfloat(stringgrid2.Cells[9,line1_2],3);
```

```
  inc (line1_2);  
  sof:=1;  
  inc (line1);  
  if line1=2 then  
  begin  
    line1_2:=1;  
    line1:=1;  
    line:=1;  
    sof2:=1;  
    for sof2:=1 to 9 do  
    begin  
      for line:=1 to 2 do  
        begin  
          Stringgrid2.Cells[sof2,line]:=' '  
        end  
      end  
    end;  
    edit5.Text:='M';  
    button7.Click;  
  end;
```

//1st key character

```

procedure TForm1.Timer3Timer(Sender: TObject);
begin
    label9.Caption:=timetostr(time)+' '+datetostr(date);
end;

```

```

procedure TForm1.Button6Click(Sender: TObject);
begin
    qdata3.Close;
    qdata3.SQL.Clear;
    qdata3.SQL.Add('select * from monitoring where no>-1 and detect =3 and
    date(waktu)=date(:waktu3)');
    qdata3.ParamByName('waktu3').AsDate:=datetimepicker1.Date;
    qdata3.ExecSQL;
    qdata3.Open;
    rvproject3.projectfile:='prin3.rav';//nama file slave3 pada MySQL
    rvproject3.Execute;
end;
{rvproject sebagai penampil data yang tersimpan pada MySQL}

```

```

{=====
procedure tambah untuk menyimpan data pada MYSQL. dari Delphi
keMySQL>>tambah,tambah1,tambah2
=====}

```

```

procedure TForm1.tambah(suhu1, kel1, tek1: double;detect:integer);
begin
    q1.Close;
    q1.SQL.Clear;
    q1.SQL.Add('insert          into          monitoring
    (waktu,suhu1,kelembaban1,tekanan1,detect) values(:wk,:t1,:rh1,:p1,:ct)');
    q1.ParamByName('t1').AsFloat:=suhu1;
    q1.ParamByName('p1').AsFloat:=tek1;
    q1.ParamByName('rh1').AsFloat:=kel1;
    q1.ParamByName('wk').AsDate:=now;
    q1.ParamByName('ct').AsInteger:=detect;
    q1.ExecSQL;
    qdata1.Refresh;
end;

```

```

procedure TForm1.tambah2(suhu2, kel2, tek2: double;detect:integer);
begin
  q2.Close;
  q2.SQL.Clear;
  q2.SQL.Add('insert                into                monitoring
(waktu,suhu2,kelembaban2,tekanan2,detect) values(:wk,:t2,:rh2,:p2,:ct)');
  q2.ParamByName('t2').AsFloat:=suhu2;
  q2.ParamByName('p2').AsFloat:=tek2;
  q2.ParamByName('rh2').AsFloat:=kel2;
  q2.ParamByName('wk').AsDateTime:=now;
  q2.ParamByName('ct').AsInteger:=detect;
  q2.ExecSQL;
  qdata2.Refresh;
end;

```

```

procedure TForm1.tambah3(suhu3, kel3, tek3: double;detect:integer);
begin
  q3.Close;
  q3.SQL.Clear;
  q3.SQL.Add('insert                into                monitoring
(waktu,suhu3,kelembaban3,tekanan3,detect) values(:wk,:t3,:rh3,:p3,:ct)');
  q3.ParamByName('t3').AsFloat:=suhu3;
  q3.ParamByName('p3').AsFloat:=tek3;
  q3.ParamByName('rh3').AsFloat:=kel3;
  q3.ParamByName('wk').AsDateTime:=now;
  q3.ParamByName('ct').AsInteger:=detect;
  q3.ExecSQL;
  qdata3.Refresh;
end;

```

```

{=====
  procedure berkaitan dengan MYSQL. Untuk membuka kembali data yang tersimpan.
  =====}

```

```

procedure TForm1.DateTimePicker1Change(Sender: TObject);
begin
  qdata1.Close;
  qdata1.SQL.Clear;
  qdata1.SQL.Add('select * from monitoring where no>-1 and detect =1 and
date(waktu)=date(:waktu1)');
  qdata1.ParamByName('waktu1').AsDate:=datetimepicker1.Date;
  qdata1.ExecSQL;
  qdata1.Open;

```

```
qdata2.Close;
qdata2.SQL.Clear;
qdata2.SQL.Add('select * from monitoring where no>-1 and detect =2 and
date(waktu)=date(:waktu2)');
qdata2.ParamByName('waktu2').AsDate:=datetimepicker1.Date;
qdata2.ExecSQL;
qdata2.Open;
qdata3.Close;
qdata3.SQL.Clear;
qdata3.SQL.Add('select * from monitoring where no>-1 and detect =3 and
date(waktu)=date(:waktu3)');
qdata3.ParamByName('waktu3').AsDate:=datetimepicker1.Date;
qdata3.ExecSQL;
qdata3.Open;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
qdata1.Close;           //seeing the saved data
qdata1.SQL.Clear;
qdata1.SQL.Add('select * from monitoring where no>-1 and detect =1 and
date(waktu)=date(:waktu1)');
qdata1.ParamByName('waktu1').AsDate:=datetimepicker1.Date;
qdata1.ExecSQL;
qdata1.Open;
rvproject1.projectfile:='prin1.rav';//inspired by someone @Fisika 05
rvproject1.Execute;
end;
```

```
procedure TForm1.Button7Click(Sender: TObject);
begin
comport1.Connected:=true;
comport1.SetRTS(false);
comport1.TransmitChar('S'); //start adc on PIC16F876
button1.Click;
end;
end.
```

{sebagai software database dipergunakan MySQL Delphi}

2. Source Code Slave/Mikrokontroler(Assembly)

```
LIST P=PIC16F876
#include "p16F876.inc"
;SOFTWARE MULTIDROP-RS485
;25 NOVEMBER 2007
;M.Dimas Khoirul Fahmi
;Pembimbing:Dr. Eng Didik R Santoso, M.Si
;=====
```

INDF	EQU	00H	
STATUS	EQU	03H	
RP0	EQU	05H	
RP1	EQU	06H	
ADCON0	EQU	1FH	
ADCON1	EQU	9FH	
ADRESL	EQU	9EH	
ADRESH	EQU	1EH	
TXREG	EQU	19H	
TXSTA	EQU	98H	
TRISA	EQU	85H	
PORTA	EQU	05H	
TRISB	EQU	86H	
PORTB	EQU	06H	
TRISC	EQU	87H	
PORTC	EQU	07H	
FSR	EQU	04H	
CNTAD	EQU	20H	
GUDANG	EQU	21H	;register hasil konversi ADC
GUDANG1	EQU	22H	
GUDANG2	EQU	23H	
GUDANG3	EQU	24H	
GUDANG4	EQU	25H	
GUDANG5	EQU	26H	
SUM1	EQU	27H	
SUM2	EQU	28H	
SUMINI	EQU	29H	;register penyimpanan karakter start adc

ORG 0X0000

init ;INISIALISASI

bsf STATUS,5
movlw h'00'
movwf TRISB
bcf STATUS,5

bsf STATUS,5
movlw D'129'
movwf SPBRG
movlw 0x80 ;RIGHT justify
movwf ADCON1 ;ADC
clrf TXSTA
BSF TXSTA,BRGH
bsf TXSTA,TXEN
bcf STATUS,5
bsf RCSTA,SPEN
bsf RCSTA,CREN
CLRF PIR1

main

CALL TERIMA
bcf STATUS,Z
MOVWF SUMINI ;register start adc

xorlw 'S' ;ADC START TOGETHER
btfsc STATUS,Z
goto adcs

bcf STATUS,Z
MOVWF SUMINI
xorlw 'P' ;KEY CHARACTER SLAVE,CHANGEABLE
btfss STATUS,Z
goto kirimgudang
goto main

adcs

BCF STATUS,RP0 ;BANK0
MOVLW 0X81 ;CH0
CALL ADC
MOVWF ADRESH
MOVWF GUDANG
BSF STATUS,RP0 ;BANK1
MOVWF ADRESL

```
BCF STATUS,RP0 ;BANK0
MOVWF GUDANG1
```

```
bcf STATUS,RP0 ;BANK0
movlw 0x89 ;ch1
call ADC
movfw ADRESH
movwf GUDANG2
bsf STATUS, RP0 ;BANK1
movfw ADRESL
bcf STATUS,RP0 ;BANK0
movwf GUDANG3
```

```
bcf STATUS,RP0 ;BANK0
movlw 0x91 ;ch2
call ADC
movfw ADRESH
movwf GUDANG4
bsf STATUS, RP0 ;BANK1
movfw ADRESL
bcf STATUS,RP0 ;BANK0
movwf GUDANG5
```

```
Goto main
```

Kirimgudang

```
CALL DELAY
CALL DELAY
CALL DELAY
BCF STATUS,RP0 ;BANK0
MOVFW GUDANG
```

```
CALL KIRIM
CALL DELAY
```

```
BCF STATUS,RP0 ;BANK0
MOVFW GUDANG1
CALL KIRIM
CALL DELAY
```

```
bcf STATUS,RP0 ;BANK0
MOVFW GUDANG2
CALL KIRIM
```

```
CALL    DELAY
bcf     STATUS,RP0    ;BANK0
MOVFW   GUDANG3
CALL    KIRIM
CALL    DELAY
```

```
bcf     STATUS,RP0    ;BANK0
MOVFW   GUDANG4
CALL    KIRIM
CALL    DELAY
bcf     STATUS,RP0    ;BANK0
MOVFW   GUDANG5
CALL    KIRIM
CALL    DELAY
```

```
GOTO    main
```

ADC:

```
MOVWF   ADCON0
CALL    DELAY
BSF     ADCON0,2      ;START CONVERT
BTFSC   ADCON0,2      ;IS IT DONE?
GOTO    $-1           ;WAIT UNTILL DONE
RETURN
```

;TRANSMITTER

KIRIM:

```
BCF     STATUS,RP0
BSF     PORTB,1       ;485transmit mode
MOVWF   TXREG
```

wait

```
BSF     STATUS,RP0
movlw   TXSTA
movwf   FSR
btfss   INDF,1
goto    wait
return
```

;RECEIVER

TERIMA:

BCF STATUS,RP0

BCF PORTB,1 ;485RECEIVE mode

btfs PIR1, RCIF

goto \$-1

movfw RCREG

return

DELAY:

movlw h'FF'

movwf SUM1

tunda0

movlw h'FF'

movwf SUM2

tunda1

decfsz SUM2

goto tunda1

decfsz SUM1

goto tunda0

return

end



Lampiran 3

Data pengujian *software* kelembaban dan suhu

3.1 Data pengujian software kelembaban

Slave1		Slave2		Slave3	
RH(%)	RHDigital(%)	RH(%)	RHDigital(%)	RH(%)	RHDigital(%)
72	71	71	70	68	68
72	71	71	70	68	68
72	71	71	70	68	68
72	71	71	69	68	68
72	71	71	69	68	68
72	71	70	69	68	68
72	71	70	69	68	68
72	71	70	69	68	68
72	71	70	69	68	68
72	71	70	69	68	68
71	71	70	69	68	68
71	71	70	69	68	68
71	71	70	69	68	68
71	71	70	69	68	68
71	71	70	69	68	68
71	71	69	69	68	68
71	70	69	69	68	68
71	70	69	69	68	68
71	70	69	68	68	68
71	70	69	68	68	68
71	70	69	68	68	68
71	69	69	68	68	68
71	69	69	68	68	68
71	69	69	68	68	68
70	69	69	68	68	68
70	69	69	68	68	68
70	69	69	70	68	68
70	68	69	70	68	68
70	68	69	70	68	68
70	68	69	69	68	68

Slave1		Slave2		Slave3	
RH(%)	RHDigital(%)	RH(%)	RHDigital(%)	RH(%)	RHDigital(%)
70	68	69	68	68	68
69	68	69	68	68	68
69	68	69	68	68	68
69	67	69	68	68	68
69	67	69	68	68	67
69	67	69	68	68	67
69	67	69	68	68	67
69	67	69	68	68	67
69	67	69	68	68	67
68	66	69	68	68	67
68	66	69	68	68	67
68	66	69	68	68	67
68	66	69	68	67	67
68	66	69	68	67	67
68	66	69	68	67	67
68	66	69	68	67	67
68	65	69	68	67	67
67	65	69	68	67	67
67	65	69	68	67	67
67	65	69	68	67	67
67	65	69	68	67	67
67	65	69	68	67	67
67	65	69	68	67	67
66	65	69	68	67	67
66	64	69	68	67	67
66	64	69	68	67	67
66	64	69	68	67	67
66	64	69	68	67	67
65	64	69	68	67	67
65	64	69	68	67	67
65	64	69	68	67	67
65	64	68	68	67	67
65	64	68	68	67	67

Slave1		Slave2		Slave3	
RH(%)	RHDigital(%)	RH(%)	RHDigital(%)	RH(%)	RHDigital(%)
65	64	68	68	67	67
65	63	68	67	67	67
65	63	68	67	67	67
65	63	68	67	67	67
65	63	68	67	67	67
64	63	68	67	67	67
64	63	68	67	67	67
64	63	68	67	67	67
64	63	68	67	67	67
64	63	68	67	67	67
64	63	68	67	67	66
64	63	68	67	67	66
64	63	68	67	67	66
64	63	68	67	67	66
64	63	67	67	67	66
64	63	67	67	67	66
64	63	67	67	67	66
64	63	67	67	67	66
64	63	67	67	67	66
64	63	67	66	67	66
64	63	67	66	67	66
64	63	67	66	67	66
64	63	67	66	67	66
64	63	67	66	67	65
64	63	67	66	67	65
64	63	67	66	67	65
64	63	67	66	67	65
64	63	67	66	67	65
64	63	67	66	67	65
64	63	67	66	67	65
64	62	67	66	67	65

Slave1		Slave2		Slave3	
RH(%)	RHDigital(%)	RH(%)	RHDigital(%)	RH(%)	RHDigital(%)
64	62	67	66	67	65
64	62	67	66	67	65
64	62	67	66	67	65
64	62	66	66	67	65
64	62	66	66	67	65
64	62	66	66	67	65
64	62	66	66	67	65
64	62	66	66	67	65
64	62	66	66	66	65
64	62	66	66	66	65
64	62	66	66	66	65
64	62	66	65	66	65
64	62	66	65	66	65
64	62	66	65	66	65
64	62	66	65	66	65
64	62	66	65	66	65
64	62	65	65	66	65
64	62	65	65	66	65
64	62	65	65	66	65
64	62	65	65	66	65
64	62	65	65	66	65
64	62	65	65	66	64
63	62	65	64	66	64
63	62	65	64	66	64
63	62	65	64	66	64
63	62	65	64	66	64
63	62	65	64	66	64
63	62	65	64	65	64
63	62	65	64	65	64
63	61	65	63	65	64
63	61	65	63	65	64

[illegible]

[illegible][illegible]

[illegible]

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,3
25,8	25,8	25,8	25,8	26,3	26,2
25,8	25,8	25,8	25,8	26,3	26,2
25,8	25,8	25,8	25,8	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,8	25,7	26,2	26,2
25,8	25,8	25,7	25,7	26,2	26,2
25,8	25,8	25,7	25,7	26,2	26,2
25,8	25,8	25,7	25,7	26,2	26,2
25,8	25,8	25,7	25,7	26,2	26
25,8	25,8	25,7	25,7	26,2	26
25,8	25,7	25,7	25,7	26,1	26
25,8	25,7	25,7	25,7	26,1	26
25,8	25,7	25,7	25,7	26,1	26
25,8	25,7	25,7	25,7	26,1	26
25,8	25,7	25,7	25,7	26,1	26
25,7	25,7	25,7	25,7	26,1	26

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
25,7	25,7	25,7	25,7	26,1	26
25,7	25,7	25,7	25,7	26	26
25,7	25,7	25,7	25,7	26	26
25,7	25,7	25,7	25,7	26	26
25,7	25,7	25,7	25,7	26	26
25,7	25,7	25,7	25,7	26	26
25,7	25,7	25,7	25,7	26	26
25,7	25,7	25,7	25,6	26	26
25,7	25,7	25,7	25,6	26	26
25,7	25,7	25,7	25,6	26	26
25,7	25,7	25,7	25,6	26	26
25,7	25,7	25,7	25,6	26	26
25,7	25,7	25,7	25,6	26	26
25,7	25,7	25,6	25,6	26	26
25,7	25,7	25,6	25,6	26	26
25,7	25,7	25,6	25,6	25,9	26
25,7	25,6	25,6	25,6	25,9	25,9
25,7	25,6	25,6	25,6	25,9	25,9
25,7	25,6	25,6	25,6	25,9	25,9
25,7	25,6	25,6	25,6	25,9	25,9
25,7	25,6	25,6	25,6	25,9	25,9
25,7	25,6	25,6	25,6	25,9	25,9
25,7	25,6	25,6	25,6	25,9	25,9
25,6	25,6	25,6	25,6	25,9	25,9
25,6	25,6	25,6	25,6	25,9	25,9
25,6	25,6	25,6	25,6	25,9	25,9
25,6	25,6	25,6	25,5	25,9	25,9
25,6	25,6	25,6	25,5	25,8	25,8
25,6	25,6	25,6	25,5	25,8	25,8
25,6	25,6	25,6	25,5	25,8	25,8

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
25,6	25,6	25,6	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,8	25,8
25,6	25,6	25,5	25,5	25,7	25,7
25,5	25,6	25,5	25,5	25,7	25,7
25,5	25,6	25,5	25,5	25,7	25,7
25,5	25,6	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,5	25,4	25,7	25,7
25,5	25,5	25,4	25,4	25,7	25,7
25,5	25,5	25,4	25,4	25,7	25,7
25,5	25,5	25,4	25,4	25,7	25,7
25,5	25,5	25,4	25,4	25,7	25,7
25,5	25,5	25,4	25,4	25,7	25,7
25,5	25,4	25,4	25,4	25,7	25,7
25,5	25,4	25,4	25,4	25,7	25,7
25,5	25,4	25,4	25,4	25,7	25,7
25,5	25,4	25,4	25,4	25,7	25,7
25,4	25,4	25,4	25,4	25,6	25,6
25,4	25,4	25,4	25,4	25,6	25,6
25,4	25,4	25,4	25,4	25,6	25,6
25,4	25,4	25,4	25,4	25,6	25,6

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
25,4	25,4	25,4	25,3	25,6	25,6
25,4	25,4	25,4	25,3	25,6	25,6
25,4	25,4	25,4	25,3	25,6	25,6
25,4	25,4	25,4	25,3	25,6	25,6
25,4	25,4	25,4	25,3	25,6	25,6
25,4	25,4	25,3	25,3	25,6	25,6
25,4	25,4	25,3	25,3	25,6	25,6
25,4	25,4	25,3	25,3	25,6	25,6
25,4	25,4	25,3	25,3	25,6	25,6
25,4	25,4	25,3	25,3	25,6	25,6
25,4	25,4	25,3	25,3	25,6	25,6
25,4	25,3	25,3	25,3	25,6	25,6
25,4	25,3	25,3	25,3	25,6	25,5
25,4	25,3	25,3	25,3	25,6	25,5
25,4	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,3	25,3	25,3	25,5	25,5
25,3	25,2	25,3	25,3	25,5	25,5
25,3	25,2	25,3	25,3	25,5	25,5
25,3	25,2	25,3	25,2	25,5	25,5
25,3	25,2	25,3	25,2	25,5	25,5
25,3	25,2	25,2	25,2	25,5	25,5

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
25,3	25,2	25,2	25,2	25,5	25,5
25,3	25,2	25,2	25,2	25,5	25,5
25,3	25,2	25,2	25,2	25,5	25,5
25,3	25,2	25,2	25,2	25,5	25,4
25,3	25,2	25,2	25,2	25,4	25,4
25,3	25,2	25,2	25,2	25,4	25,4
25,3	25,2	25,2	25,2	25,4	25,4
25,3	25,2	25,2	25,2	25,4	25,4
25,3	25,2	25,2	25,2	25,4	25,4
25,3	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,2	25,4	25,4
25,2	25,2	25,2	25,1	25,4	25,4
25,2	25,2	25,2	25,1	25,4	25,4
25,2	25,2	25,2	25,1	25,4	25,3
25,2	25,2	25,1	25,1	25,3	25,3
25,2	25,2	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,2	25,1	25,1	25,1	25,3	25,3
25,1	25,1	25,1	25,1	25,3	25,3

[illegible]

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,9	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,9	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,9	24,8	24,8	24,8	25,1	25,1
24,8	24,8	24,8	24,8	25,1	25,1
24,8	24,8	24,8	24,8	25,1	25,1
24,8	24,8	24,8	24,8	25,1	25

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
24,8	24,8	24,8	24,7	25,1	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,8	24,7	25	25
24,8	24,8	24,7	24,7	25	25
24,8	24,8	24,7	24,7	25	25
24,8	24,8	24,7	24,7	25	25
24,8	24,8	24,7	24,7	25	25
24,8	24,8	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,7	25	25
24,8	24,7	24,7	24,6	25	25
24,8	24,7	24,7	24,6	25	25
24,8	24,7	24,7	24,6	25	25
24,8	24,7	24,7	24,6	25	25

[illegible]

Slave1		Slave2		Slave 3	
T (°C)	TDigital (°C)	T (°C)	TDigital (°C)	T (°C)	TDigital (°C)
24,6	24,6	24,6	24,6	24,9	24,9
24,6	24,6	24,6	24,6	24,9	24,9
24,6	24,6	24,6	24,6	24,9	24,9
24,6	24,6	24,6	24,6	24,9	24,9
24,6	24,6	24,6	24,6	24,9	24,9
24,6	24,6	24,6	24,6	24,9	24,9
24,6	24,6	24,6	24,6	24,9	24,9

