



UNIVERSITAS INDONESIA

**AKUISISI DATA TEMPERATUR DAN TEKANAN UDARA BERBASIS
MIKROKONTROLER H8/3069F**

SKRIPSI

NURRACHMI PUTRI UTAMI

0606068543

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

PROGRAM STUDI FISIKA

DEPOK

JUNI 2011

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : NURRACHMI PUTRI UTAMI

NPM : 0606068543

Tanda Tangan :



Tanggal : 14 Juni 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : NURRACHMI PUTRI UTAMI

NPM : 0606068543


Program Studi : Instrumentasi Elektronika S1

Judul Skripsi :

AKUISISI DATA TEMPERATUR DAN TEKANAN UDARA BERBASIS
MIKROKONTROLER H8/3069F

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Eng. Supriyanto, M.Sc ()

Penguji I : Dr. Tony Mulia ()

Penguji II : Dr. Sastra Kusuma Wijaya ()

Ditetapkan di : Depok

Tanggal : 14 Juni 2011

KATA PENGANTAR

Segala puji bagi Allah SWT, Tuhan seru sekalian alam atas limpahan rahmat, hidayah serta karunia-Nya yang tak terhingga sehingga penulis dapat menyelesaikan skripsi ini dengan baik.

Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Sains Jurusan Fisika pada Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

Penulis menyadari bahwa, selesainya skripsi ini tidak terlepas dari bantuan, bimbingan, dorongan dan doa yang tulus dari banyak pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini. Tanpa itu semua sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Untuk itu penulis ingin menyampaikan ucapan terima kasih yang setulus - tulusnya kepada :

1. (alm) ayahanda, ibunda, kakak-kakak dan adik ku atas do'a, cinta dan dukungannya yang tak berhingga yang menghidupkan semangatku untuk maju.
2. Dr. Eng. Supriyanto, M.Sc selaku dosen pembimbing yang telah membimbing penulis dalam segala hal, baik dalam ilmu pengetahuan, nasehat, dorongan, semangat serta segala fasilitas yang telah diberikan dalam menyelesaikan tugas akhir ini.
3. Ir. Moch. Ichsan selaku Ka. Unit Pengamat Wahana Dirgantara (UPWD) LAPAN Biak yang telah mengenalkan keeleganan APRS dan memberikan fasilitas yang amat nyaman selama pengerjaan skripsi di LAPAN sehingga skripsi ini dapat terselesaikan.
4. Bapak Chusnul Tri Judianto, selaku Kabid. Ruas Bumi Dirgantara LAPAN Rancabungur.
5. Dr. Tony Mulia selaku penguji I dan Dr. Sastra Kusuma Wijaya selaku penguji II atas saran dan kritiknya baik sebelum maupun sesudah penulis melakukan sidang.
6. Dosen-dosen Dept. Fisika Universitas Indonesia yang dengan sabarnya mengajari saya dan teman-teman. Jasa-mu sungguh luar biasa.....

7. Mas Deddy, Pak Unggul, Mba Ocha, Mba Nayla, Mas Ade, Mas Hadi, Mas Ronny, Pak Endro, Pak Efendi.” *Kalian semua the best deh pokoknya. Maap ya...saya sering ngerepotin kalian semua...hehehe*” (^_^)
8. Teman-teman seperjuangan dalam menyusun skripsi : Momoy dan Haris. *Thanks berat untuk kebersamaannya lho...akhirnya AHM bubar juga. Ihiyy...*
9. Pa Karno dan Ka Muchtar yang telah bersedia meluangkan waktu untuk diskusi dan bimbingan sehingga skripsi ini dapat selesai dengan baik.
10. Tim Riset H8 mulai dari sesepuh hingga angkatan muda. Ka Ilham, Ka Tyo, Ka Thea, Ka Tiwi, Ka Nurma, Ka Moku, Rizki, Lia, Faizal, Imas, Rusyda, Yulia, Nella, Fikri, Salas, Herlina, Yusuf, Faldo, Ferry, Vika, Fina, dan Cherry. *Ga bosan deh belajar H8 dan Python bareng kalian semua. Taratengkyu udah dibantuin.*
11. Instrumen dan instruwomen '06 dan semua anak fisika UI, para rangers dari second flat, pelangi, mafaza, galaksi, ikaris. *Makasih lho buat doa dan semangat dari kalian semua.*
12. *For everyone who gave me support and many kind of help :D*

Menyadari kekurangan dalam penulisan skripsi ini serta kemungkinan jauh dari sempurna, untuk itu penulis tidak menutup diri dari segala saran dan kritik yang sifatnya membangun dari semua pihak guna menyempurnakan isi skripsi ini. Akhir kata semoga skripsi ini dapat bermanfaat bagi siapa pun yang mengkajinya.

Depok, 14 Juni 2011

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

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

Nama : Nurrachmi Putri Utami
NPM : 0606068543
Program Studi : Instrumentasi Elektronika S1
Departemen : Fisika
Fakultas : Matematika dan Ilmu pengetahuan Alam
Jenis karya : Skripsi

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

**AKUISISI DATA TEMPERATUR DAN TEKANAN UDARA BERBASIS
MIKROKONTROLER H8/3069F**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*Database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 14 Juni 2011

Yang menyatakan



(Nurrachmi Putri Utami)

Nama : Nurrachmi Putri Utami
Program studi : Fisika
Judul skripsi : Akuisisi Data Temperatur dan Tekanan Udara Berbasis
Mikrokontroler H8/3069F

ABSTRAK

Rancangan alat akuisisi data temperatur dan tekanan udara berbasis mikrokontroler H8/3069F menggunakan sensor temperatur digital DS18B20, sensor tekanan udara analog MPX4155A. Sistem ini juga dilengkapi informasi waktu, posisi dan ketinggian yang diperoleh dari GPS Garmin 18 LVC. Perangkat lunak yang bekerja pada mikrokontroler dibuat menggunakan bahasa C yang dikategorikan sebagai bahasa *mid-level* yang mudah diimplementasikan pada mikrokontroler. Adapun komunikasi sensor DS18B20, sensor MPX4155A dan GPS ke mikrokontroler masing-masing menggunakan *1-wire*, ADC, dan RS-232. Selanjutnya, hasil akuisisi ditampilkan dalam bentuk *Graphical User Interface* (GUI) yang dibuat dengan bahasa pemrograman Python dan database berbasis MySQL.

Kata Kunci : ADC, DS18B20, GPS Garmin 18 LVC, mikrokontroler H8/3069F, MPX4115A, MySQL, *1-wire*, Python, RS-232.

Name : Nurrachmi Putri Utami
Program of study : Physics
Title of Essay : Data Acquisition of Temperature and Air Pressure based on Microcontroller H8/3069F

ABSTRACT

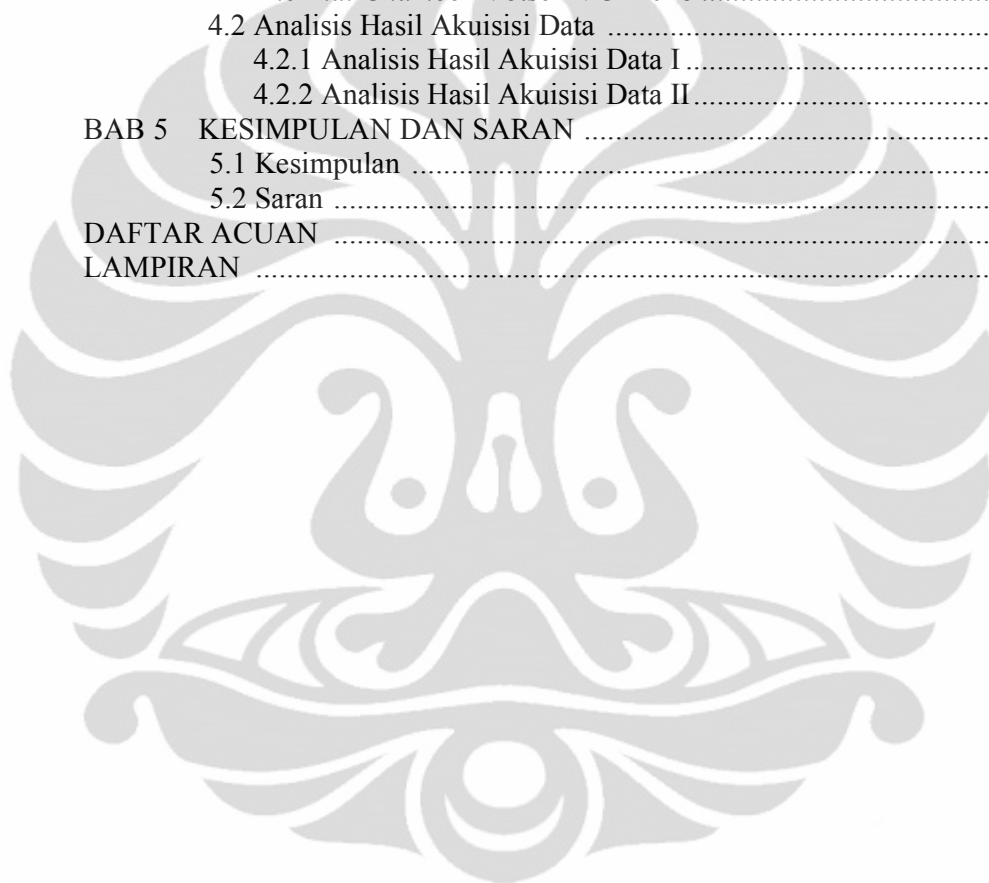
Instrument design of acquisition temperature and air pressure based on microcontroller H8/3069F using digital temperature sensor DS18B20, analog air pressure sensor MPX4155A. The system equipped with time, position, and altitude information obtained from GPS Garmin 18 LVC. The software works on microcontroller made by using C language which is categorized as middle-level language and easy to implement on a microcontroller. Interfacing of DS18B20 sensor, MPX4155A sensor and GPS to microcontroller respectively using 1-wire, ADC, and RS-232. Moreover, the result of acquisition is shown in the form of graphical user interface (GUI) created with Python programming language and database based on MySQL.

Key Words : ADC, DS18B20, GPS Garmin 18 LVC, microcontroller H8/3069F, MPX4115A, MySQL, 1-wire, Python, RS-232.

DAFTAR ISI

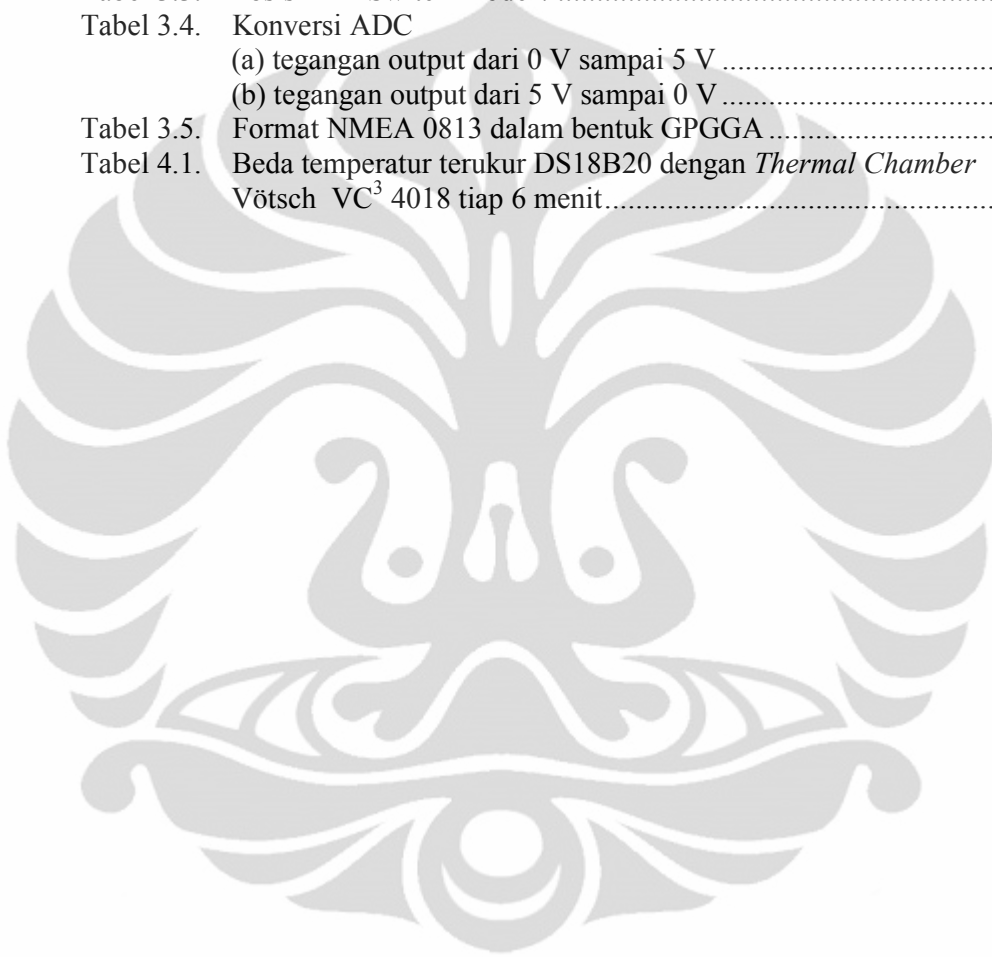
HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR.....	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
ABSTRAK.....	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah.....	1
1.3 Tujuan dan Penelitian	2
1.4 Metodologi Penelitian.....	2
1.5 Sistematika Penulisan	3
BAB 2 LANDASAN TEORI	5
2.1 Unsur Cuaca dan Iklim	5
2.1.1 Angin.....	5
2.1.2 Curah Hujan	6
2.1.3 Suhu Udara.....	6
2.1.4 Kelembaban Udara	7
2.1.5 Tekanan Udara.....	7
2.2 Sensor Temperatur Digital DS18B20	8
2.3 Sensor Tekanan Udara Analog MPX4115A	11
2.4 GPS.....	13
2.4.1 Penentuan Posisi dengan GPS	14
2.4.2 Format Data GPS	14
2.4.3 GPS Garmin 18 LVC	15
2.5 Mikrokontroler H8/300H.....	16
2.5.1 <i>Internal ROM</i>	17
2.5.2 <i>Internal RAM</i>	17
2.5.3 I/O port: 11 I/O port.....	17
2.5.4 <i>Internal A/D Converter</i> dengan Resolusi 10-bit × 8 Channel	17
2.5.5 <i>Internal 8-bit D/A Converter</i> × 2 Channel	18
BAB 3 PERANCANGAN ALAT.....	19
3.1 Perangkat Keras	19
3.1.1 Pengoperasian Sensor DS18B20	20
3.1.2 Konversi Analog ke Digital Menggunakan Mikrokontroler H8/3069F	21
3.1.3 Pengoperasian Sensor MPX4115A	23
3.1.4 Pengoperasian GPS Garmin 18 LVC	23
3.2 Perangkat Lunak	24

3.3	Pengujian Sensor Temperatur Digital DS18B20	25
3.3.1	Hasil dan Analisis Data Rom Code DS18B20	27
3.3.2	Hasil dan Analisis Data Temperatur	28
3.4	Uji Linearitas ADC Internal Mikrokontroler H8/3069F	29
3.5	Hasil dan Analisis Tekanan Udara	32
3.6	Uji Hasil dan Analisis Pengoperasian GPS Garmin 18 LVC	33
3.7	Hasil dan Analisi Integrasi Sensor dan GPS.....	34
BAB 4	ANALISA HASIL PENELITIAN	38
4.1	Hasil Perbandingan Pengukuran Temperatur DS18B20 dan <i>Thermal Chamber Vötsch VC³ 4018</i>	38
4.2	Analisis Hasil Akuisisi Data	40
4.2.1	Analisis Hasil Akuisisi Data I	41
4.2.2	Analisis Hasil Akuisisi Data II	48
BAB 5	KESIMPULAN DAN SARAN	50
5.1	Kesimpulan	50
5.2	Saran	51
	DAFTAR ACUAN	52
	LAMPIRAN	53



DAFTAR TABEL

Tabel 2.1.	Deskripsi pin DS18B20	9
Tabel 2.2.	NMEA <i>sentence</i> dalam bentuk \$GPGGA.....	16
Tabel 3.1.	Konfigurasi pin ADC internal mikrokontroler H8/3069F.....	21
Tabel 3.2.	Posisi DIP Switch Mode-3	25
Tabel 3.3.	Posisi DIP Switch Mode-7	26
Tabel 3.4.	Konversi ADC	
	(a) tegangan output dari 0 V sampai 5 V	30
	(b) tegangan output dari 5 V sampai 0 V	30
Tabel 3.5.	Format NMEA 0813 dalam bentuk GPGGA	33
Tabel 4.1.	Beda temperatur terukur DS18B20 dengan <i>Thermal Chamber</i> Vötsch VC ³ 4018 tiap 6 menit.....	39



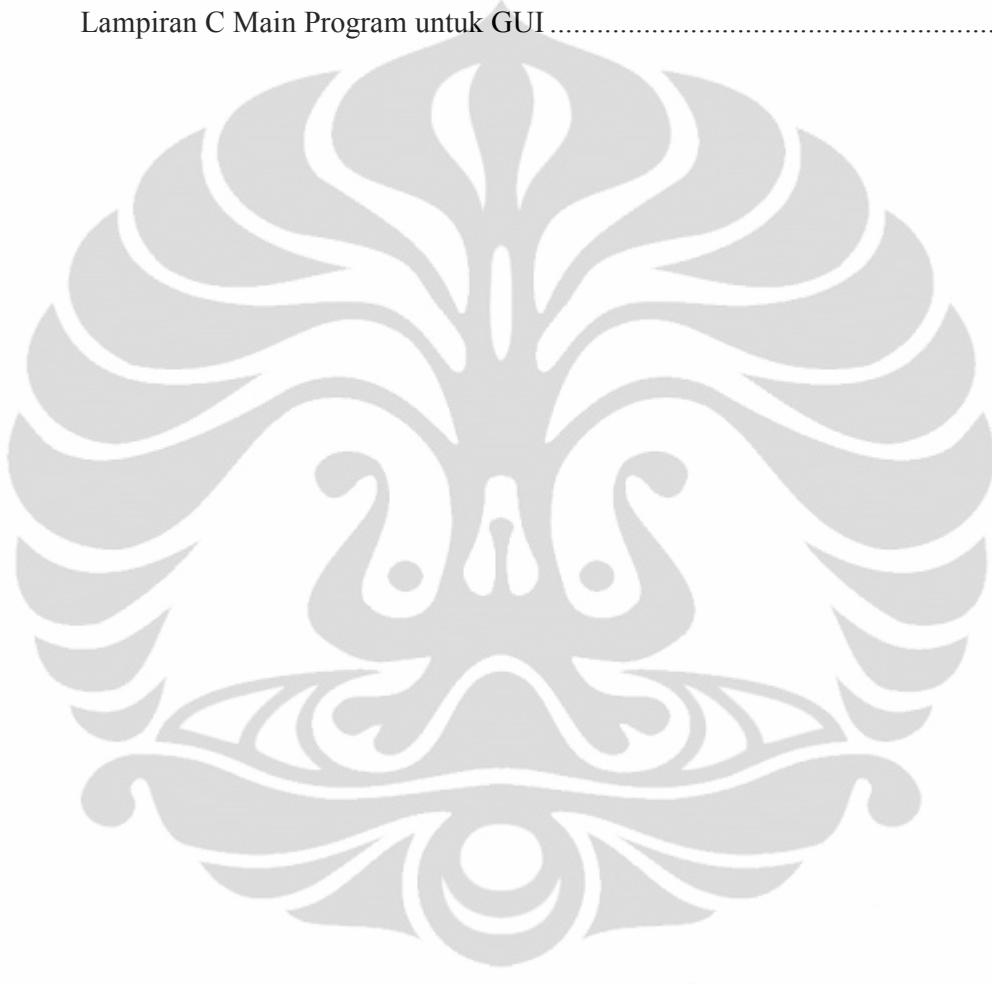
DAFTAR GAMBAR

Gambar 1.1.	Perancangan sistem perangkat keras untuk akuisisi	2
Gambar 1.2.	Diagram alir langkah-langkah penelitian	3
Gambar 2.1.	Perubahan tekanan terhadap ketinggian di atas permukaan bumi ...	8
Gambar 2.2.	Bentuk fisik DS18B20	9
Gambar 2.3.	Blok diagram DS18B20	9
Gambar 2.4.	Pensuplaian dengan <i>mode parasit power</i>	10
Gambar 2.5.	Pensuplaian dengan mode konvensional	10
Gambar 2.6.	64-bit ROM <i>code</i> dari DS18B20	11
Gambar 2.7.	Skematik sensor tekanan untuk <i>unibody package</i> dan <i>small outline package</i>	12
Gambar 2.8.	<i>Cross-Sectional Diagram</i> MPX4115A	12
Gambar 2.9.	Power supply decoupling dan filter output yang direkomendasikan untuk MPX4115A	13
Gambar 2.10.	Sinyal output MPX4115A relatif terhadap tekanan udara	13
Gambar 2.11.	GPS Garmin 18 LVC	15
Gambar 2.12.	Minimun sistem mikrokontroler H8/3069F	17
Gambar 3.1.	Perancangan sistem akuisisi data	19
Gambar 3.2.	Rangkaian operasi sensor DS18B20 ke mikrokontroler H8/3069F dengan mode <i>power supply</i> eksternal	21
Gambar 3.3.	<i>A/D control register</i> (ADCR) H8/3069F	22
Gambar 3.4.	<i>A/D control/status register</i> (ADCSR) H8/3069F	23
Gambar 3.5.	Rangkaian operasi sensor MPX4115A ke mikrokontroler H8/3069F	23
Gambar 3.6.	Interkoneksi GPS Garmin 18 LVC ke Mikrokontroler H8/3069F	24
Gambar 3.7.	Tampilan GUI sebelum di hubungkan ke mikrokontroler berbasis Python	25
Gambar 3.8.	Langkah-langkah saat melakukan <i>writing</i> program pembacaan ROM Code DS18B20 ke mikrokontroler H8/3069F	26
Gambar 3.9.	Langkah-langkah saat melakukan <i>writing</i> program pengoperasian DS18B20 ke mikrokontroler H8/3069F	27
Gambar 3.10.	<i>Family code</i> ROM DS18B20	28
Gambar 3.11.	Pengukuran suhu kamar dengan menggunakan DS18B20	29
Gambar 3.12.	Interkoneksi ADC mikrokontroler H8/3069F dengan potensiometer	29
Gambar 3.13.	Grafik perbandingan output potensiometer pada multimeter dan output ADC	31
Gambar 3.14.	Grafik perbandingan output potensiometer pada multimeter dan konversi	31
Gambar 3.15.	Nilai tekanan udara dari sensor MPX4115A	32
Gambar 3.16.	Hasil porting GPS Garmin 18 LVC ke mikrokontroler H8/3069F dengan format GPGGA	33
Gambar 3.17.	Diagram alir program utama	35
Gambar 3.18.	Diagram alir program untuk aktivasi <i>interrupt</i>	35
Gambar 3.19.	Langkah-langkah saat melakukan <i>writing</i> program integrasi sensor-sensor dan GPS	36
Gambar 3.20.	Interkoneksi awal sistem perangkat keras	36

Gambar 3.21.	Hasil porting GPS Garmin 18LVC, sensor DS18B20 dan MPX4115A ke mikrokontroler H8 untuk format GPGGA	37
Gambar 4.1.	<i>Thermal chamber</i> Vötsch VC ³ 4018.....	38
Gambar 4.2.	Hasil perbandingan pengukuran temperatur dari DS18B20 dan <i>thermal chamber</i> Vötsch VC ³ 4018.....	39
Gambar 4.3.	Peletakan perangkat keras selama akuisisi.....	41
Gambar 4.4.	Interkoneksi catu daya sistem perangkat keras.....	41
Gambar 4.5.	GUI (<i>Graphical User Interface</i>) untuk sistem akuisisi berbasis Python.....	42
Gambar 4.6.	Tampilan <i>data base</i> berbasis MySQL.....	43
Gambar 4.7.	Hasil pencitraan posisi selama akuisisi berlangsung	43
Gambar 4.8.	Grafik perubahan temperatur terhadap ketinggian	44
Gambar 4.9.	Grafik perubahan temperatur terhadap waktu dari daerah rendah ke daerah tinggi	44
Gambar 4.10.	Grafik perubahan temperatur terhadap waktu dari daerah tinggi ke daerah rendah	45
Gambar 4.11.	Grafik perubahan tekanan udara terhadap ketinggian.....	46
Gambar 4.12.	Grafik perubahan tekanan udara terhadap waktu dari daerah rendah ke daerah tinggi	47
Gambar 4.13.	Grafik perubahan tekanan udara terhadap waktu dari daerah tinggi ke daerah rendah.....	47
Gambar 4.14.	Peletakan sensor temperatur dan tekanan udara di luar rumah	48
Gambar 4.15.	Grafik perubahan temperatur terhadap waktu	49
Gambar 4.16.	Grafik perubahan tekanan udara terhadap waktu	49

DAFTAR LAMPIRAN

Lampiran A Mikrokontroler H8/3069F.....	53
Lampiran B Program C untuk mikrokontroler H8/3069F.....	56
Lampiran C Main Program untuk GUI.....	60



BAB I

PENDAHULUAN

1.1 Latar Belakang

Cuaca dan iklim merupakan akibat dari proses-proses yang terjadi di atmosfer yang menyelubungi bumi. Cuaca dan iklim adalah suatu keadaan yang terjadi di permukaan bumi yang dipengaruhi oleh kondisi udara, yaitu tekanan dan temperatur [1]. Cuaca dan iklim merupakan gejala alamiah yang sangat penting bagi kehidupan manusia, dengan mengetahui pola cuaca dan iklim seperti periode musim hujan dan kemarau, maka para petani dapat menentukan musim tanam yang tepat agar produksi pertaniannya baik. Sektor-sektor kehidupan lainnya yang berkaitan dengan kondisi cuaca dan iklim, seperti pada bidang perikanan, transportasi, perkebunan, hingga pariwisata.

Seiring perkembangan teknologi, lahirlah stasiun cuaca yang dapat ditempatkan di suatu tempat tertentu. Umumnya stasiun cuaca ini tidak bergerak, sehingga memerlukan biaya yang cukup besar untuk meletakkan stasiun cuaca berupa AWS (*Automatic Weather System*) di berbagai titik dengan ketinggian yang berbeda.

Oleh karena itu, dilakukan penelitian untuk pemantauan parameter cuaca secara *mobile* pada ketinggian yang berbeda. Adapun akuisisi data yang dilakukan adalah berupa parameter cuaca, seperti temperatur dan tekanan udara. Pusat kontrol sistem akuisisi data temperatur diatur oleh mikrokontroler H8/3069F. Mikrokontroler ini memiliki performa, seperti tersedianya 11 I/O port, 8 channel ADC 10 bit, *serial communication interface*, dan fitur-fitur lainnya. Selain itu, mikrokontroler H8/3069F membutuhkan konsumsi power yang rendah [2].

1.2 Batasan Masalah

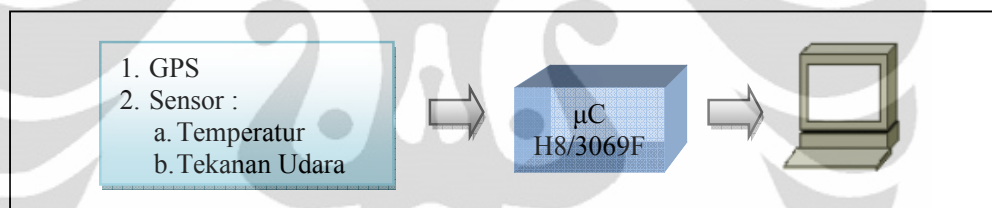
Pada penelitian ini, data yang akan diakuisisi berupa parameter temperatur dan tekanan udara. Keseluruhan data ditampilkan ke PC dengan GUI (*Graphical User Interface*) berbasis Python dan dilengkapi penyimpanan data dengan *data base* berbasis MySQL.

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mencari hubungan suhu dan tekanan udara terhadap ketinggian dengan memindahkan perangkat keras sistem akuisisi data dari daerah yang rendah ke daerah yang tinggi dan sebaliknya.

1.4 Metodologi Penelitian

Perancangan penelitian sistem akuisisi temperatur dan tekanan udara, dimulai dengan mempelajari penggunaan sensor dan GPS yang akan digunakan untuk diprogram ke mikrokontroler H8/3069F dengan pemrograman bahasa C. Selanjutnya, hasil pengambilan data akan ditampilkan dengan GUI (*Graphic User Interface*) yang menggunakan bahasa pemrograman Python. Sedangkan penyimpanan data disimpan dalam *data base* berbasis MySQL.



Gambar 1.1. Perancangan sistem perangkat keras untuk akuisisi

Metode penelitian yang akan dilakukan terdiri dari beberapa tahap antara lain :

a. Studi kepustakaan

Studi kepustakaan dilakukan untuk mengumpulkan informasi dan pustaka yang berkaitan dengan penelitian ini baik dari literatur, internet, buku-buku, serta penjelasan informal dari pihak terkait dan lainnya.

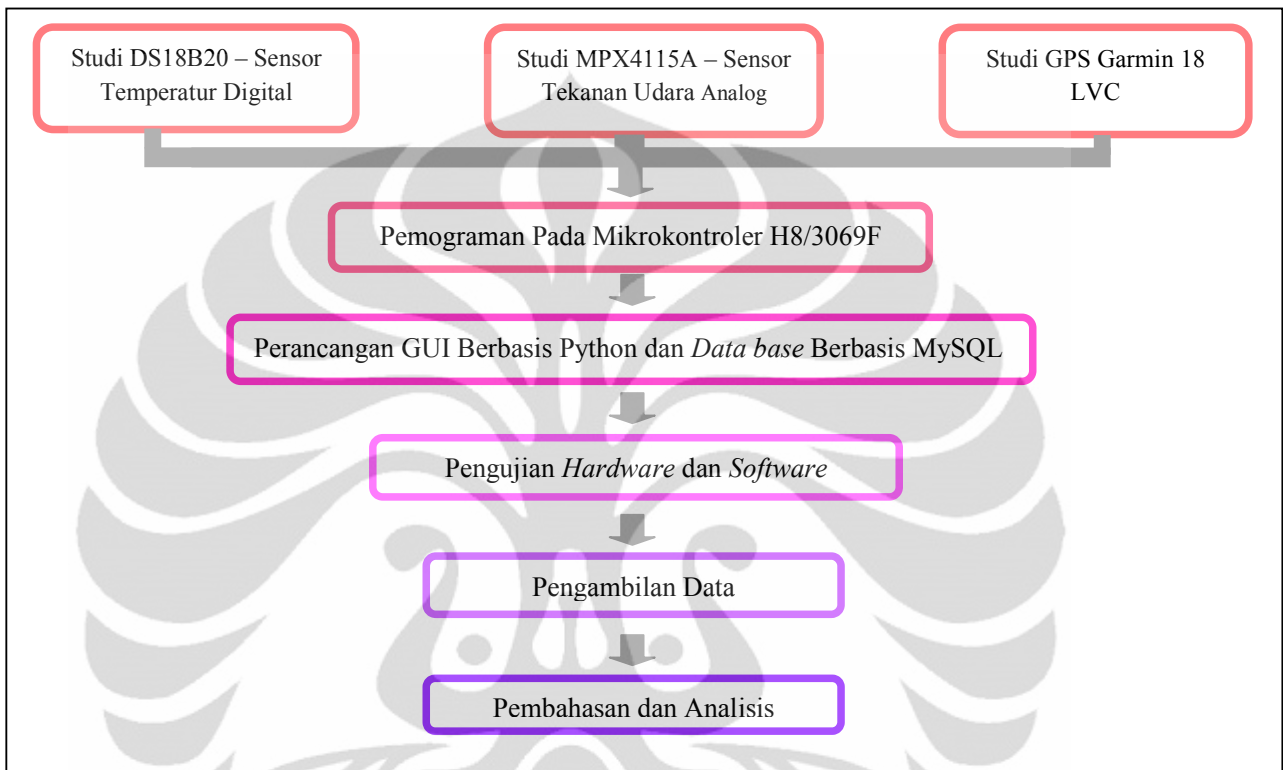
b. Studi peralatan

Studi peralatan bertujuan mempelajari karakteristik dan spesifikasi alat yang akan digunakan sehingga diperoleh pembelajaran yang tepat tentang alat yang akan dipakai.

c. Penelitian laboratorium

Penelitian laboratorium dilakukan untuk merakit alat, pengujian alat dan menganalisis alat yang telah dibuat.

Berikut ini adalah langkah-langkah penelitian :



Gambar 1.2. Diagram alir langkah-langkah penelitian

1.5 Sistematika Penulisan

Sistematika penulisan skripsi terdiri atas lima bab yang secara garis besar dapat diuraikan sebagai berikut :

1. Bab 1 Pendahuluan

Bab ini memuat tentang latar belakang dari penelitian, tujuan, metode yang digunakan, dan juga pembatasan masalah pada penelitian yang dilakukan.

2. Bab 2 Landasan Teori

Bab ini memuat secara garis besar teori dasar yang berhubungan dengan penelitian.

3. Bab 3 Perancangan Alat

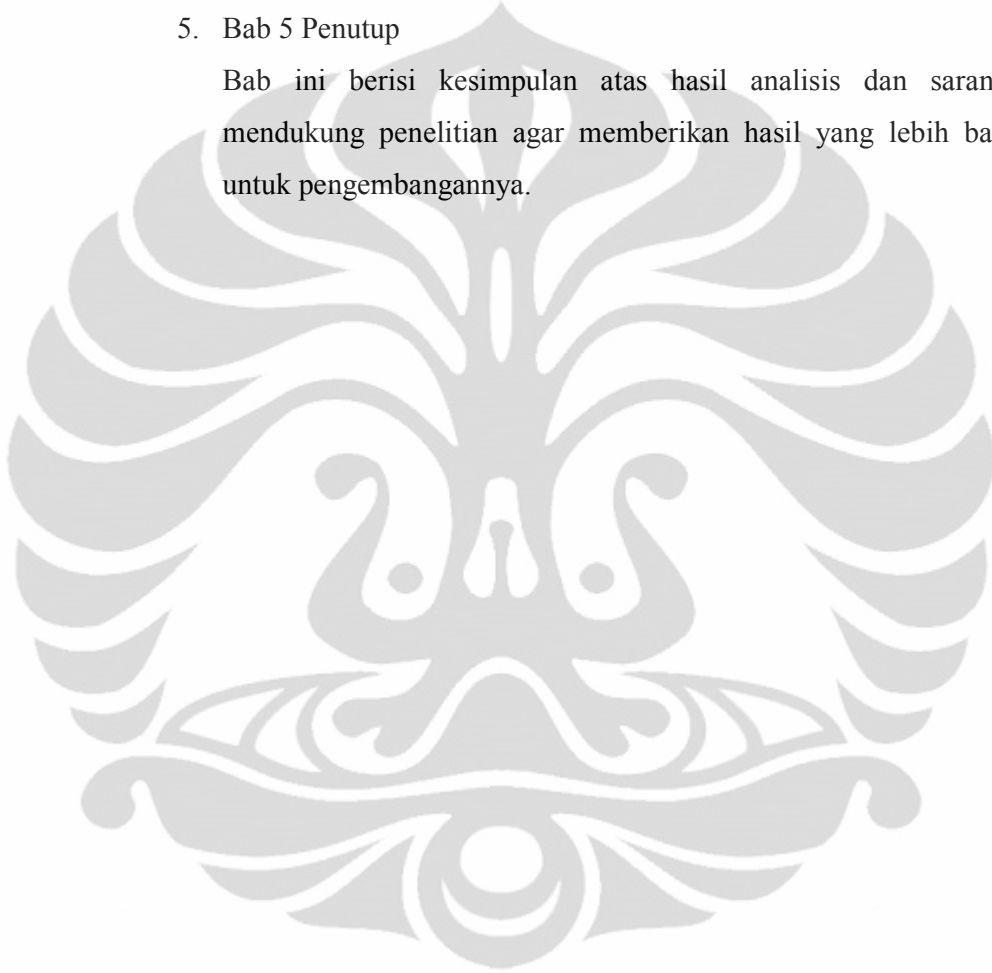
Bab ini memuat perancangan perangkat keras dan perangkat lunak

4. Bab 4 Analisis Hasil Penelitian

Bab ini berisi penjelasan mengenai hasil penelitian dan analisis dari sistem yang telah dibuat.

5. Bab 5 Penutup

Bab ini berisi kesimpulan atas hasil analisis dan saran yang mendukung penelitian agar memberikan hasil yang lebih baik lagi untuk pengembangannya.



BAB 2

LANDASAN TEORI

2.1 Unsur Cuaca dan Iklim

Cuaca adalah keadaan udara pada saat tertentu dan di wilayah tertentu yang relatif sempit dan pada jangka waktu yang singkat. Cuaca itu terbentuk dari gabungan unsur cuaca dan jangka waktu cuaca bisa hanya beberapa jam saja. Misalnya: pagi hari, siang hari atau sore hari, dan keadaannya bisa berbeda-beda untuk setiap tempat serta setiap jamnya [3].

Sedangkan iklim adalah keadaan cuaca rata-rata dalam waktu satu tahun yang penyelidikannya dilakukan dalam waktu yang lama (\pm minimal 30 tahun) dan meliputi wilayah yang luas [3]. Iklim dapat terbentuk karena adanya:

- a. Rotasi dan revolusi bumi sehingga terjadi pergeseran semu harian matahari dan tahunan.
- b. Perbedaan lintang geografi dan lingkungan fisis. Perbedaan ini menyebabkan timbulnya penyerapan panas matahari oleh bumi sehingga besar pengaruhnya terhadap kehidupan di bumi.

Ada beberapa unsur yang mempengaruhi keadaan cuaca dan iklim suatu daerah atau wilayah, yaitu: suhu atau temperatur udara, tekanan udara, angin, kelembaban udara, dan curah hujan [3].

2.1.1 Angin

Angin adalah udara yang bergerak dari daerah bertekanan udara tinggi ke daerah bertekanan udara rendah [3]. Kecepatan angin dapat ditentukan oleh beberapa faktor, antara lain:

- a. Besar kecilnya gradien barometrik [3]

Gradien Barometrik, yaitu angka yang menunjukkan perbedaan tekanan udara melalui dua garis isobar pada garis lurus. Menurut hukum Stevenson bahwa kecepatan angin bertiup berbanding lurus dengan gradien barometriknya. Semakin besar gradien barometriknya, semakin besar pula kecepatannya.

b. Relief Permukaan Bumi [3]

Angin bertiup kencang pada daerah yang reliefnya rata dan tidak ada rintangan. Sebaliknya bila bertiup pada daerah yang reliefnya besar dan rintangannya banyak, maka angin akan berkurang kecepatannya.

c. Ada Tidaknya Tumbuh-tumbuhan [3]

Banyaknya pohon-pohonan akan menghambat kecepatan angin dan sebaliknya, bila pohon-pohonannya jarang maka sedikit sekali member hambatan pada kecepatan angin.

d. Tinggi dari Permukaan Tanah [3]

Angin yang bertiup dekat dengan permukaan bumi akan mendapatkan hambatan karena bergesekan dengan muka bumi, sedangkan angin yang bertiup jauh di atas permukaan bumi bebas dari hambatan-hambatan.

2.1.2 Curah Hujan

Curah hujan adalah jumlah air hujan yang turun pada suatu daerah dalam waktu tertentu. Alat untuk mengukur banyaknya curah hujan disebut *Rain Gauge*. Curah hujan diukur dalam harian, bulanan, dan tahunan. Curah hujan yang jatuh di wilayah Indonesia dipengaruhi oleh beberapa faktor antara lain :

1. Bentuk medan atau topografi.
2. Arah lereng medan.
3. Arah angin yang sejajar dengan garis pantai.
4. Jarak perjalanan angin di atas medan datar.

2.1.3 Suhu Udara

Suhu udara adalah keadaan panas atau dinginnya udara [3]. Alat untuk mengukur suhu udara atau derajat panas disebut termometer. Biasanya pengukuran dinyatakan dalam skala Celsius (C), Reamur (R), dan Fahrenheit (F). Suhu udara tertinggi di muka bumi adalah di daerah tropis (sekitar ekuator) dan makin ke kutub, makin dingin.

Di lain pihak, pada waktu kita mendaki gunung, suhu udara terasa dingin jika ketinggian bertambah. Tiap kenaikan bertambah 100 meter, suhu udara berkurang (turun) rata-rata $0,6^{\circ}\text{C}$. Penurunan suhu semacam ini disebut gradien temperatur vertikal atau *lapse rate*. Pada udara kering, besar lapse rate adalah 1°C

[3]. Faktor-faktor yang mempengaruhi tinggi rendahnya suhu udara suatu daerah adalah :

- b. Lama penyinaran matahari.
- c. Sudut datang sinar matahari.
- d. Relief permukaan bumi.
- e. Banyak sedikitnya awan.

2.1.4 Kelembaban Udara

Di udara terdapat uap air yang berasal dari penguapan samudra (sumber yang utama). Sumber lainnya berasal dari danau-danau, sungai-sungai, tumbuh-tumbuhan, dan sebagainya. Makin tinggi suhu udara, makin banyak uap air yang dapat dikandungnya. Hal ini berarti makin lembablah udara tersebut. Alat untuk mengukur kelembaban udara dinamakan hygrometer atau psychrometer [3].

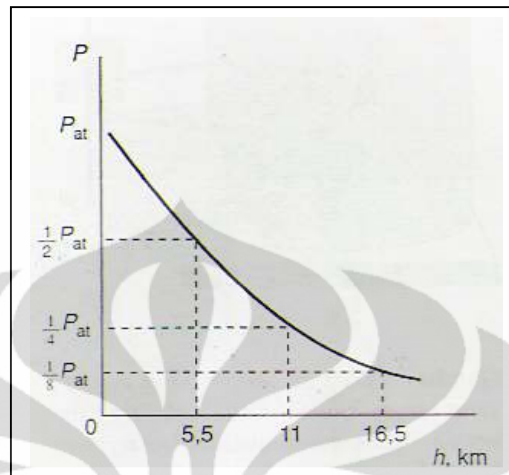
2.1.5 Tekanan Udara

Tekanan udara adalah suatu gaya yang timbul akibat adanya berat dari lapisan udara. Besarnya tekanan udara disetiap tempat pada suatu saat berubah-ubah. Makin tinggi suatu tempat dari permukaan laut, makin rendah tekanan udaranya. Hal ini disebabkan karena makin berkurangnya udara yang menekan. Besarnya tekanan udara diukur dengan barometer [3].

Tekanan udara dapat dibedakan menjadi 3 macam [3], yaitu:

1. Tekanan udara tinggi, lebih dari 1013 mb.
2. Tekanan udara rendah, kurang dari 1013 mb.
3. Tekanan di permukaan laut, sama dengan 1013 mb.

Pada gambar 2.1 menunjukkan jenis penurunan tekanan udara secara eksponensial. Pada ketinggian sekitar 5.5 km (18.000 ft), besar tekanan udara separo nilainya pada ketinggian laut. Jika kita naik lagi 5.5 km sampai ketinggian 11 km (suatu ketinggian khas untuk pesawat terbang), tekanan kembali diparo sehingga menjadi seperempat nilainya pada ketinggian laut, dan seterusnya [4].



Gambar 2.1 Perubahan tekanan terhadap ketinggian di atas permukaan bumi

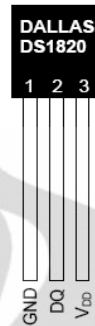
2.2 Sensor Temperatur Digital DS18B20

DS18B20 adalah sensor temperatur digital yang menyediakan 9 bit hingga 12 bit untuk pengukuran temperatur dalam celcius. Komunikasi sensor ini melalui *1-wire bus* yang berarti hanya membutuhkan satu jalur data untuk berkomunikasi dengan mikrokontroler. Sensor ini beroperasi untuk rentang temperature -55°C hingga $+125^{\circ}\text{C}$ dan memiliki keakurasian $\pm 0.5^{\circ}\text{C}$ pada rentang -10°C hingga $+85^{\circ}\text{C}$. Tiap sensor DS18B20 memiliki kode serial 64-bit yang unik, yang memiliki fungsi *1-wire bus* yang sama. Sehingga memungkinkan untuk menggunakan banyak sensor DS18B20 yang terdistribusi pada suatu area. Aplikasi ini menguntungkan untuk sistem pemantauan temperatur [5].

Bentuk fisik DS18B20 ditunjukkan oleh Gambar 2.1 dengan spesifikasi sebagai berikut :

- Interface sensor berupa *1-wire* yang hanya membutuhkan satu pin dalam satu port untuk komunikasi.
- Memiliki kode serial 64-bit yang unik yang tersimpan di *on-board* ROM.
- Rentang *Power supply* 3 V sampai 5,5 V.
- Pengukuran temperatur dari -55°C hingga $+125^{\circ}\text{C}$.
- Memiliki keakurasian $\pm 0.5^{\circ}\text{C}$ pada rentang -10°C hingga $+85^{\circ}\text{C}$.

- f. Memiliki resolusi thermometer yang dapat dipilih mulai dari 9 bit hingga 12 bit.

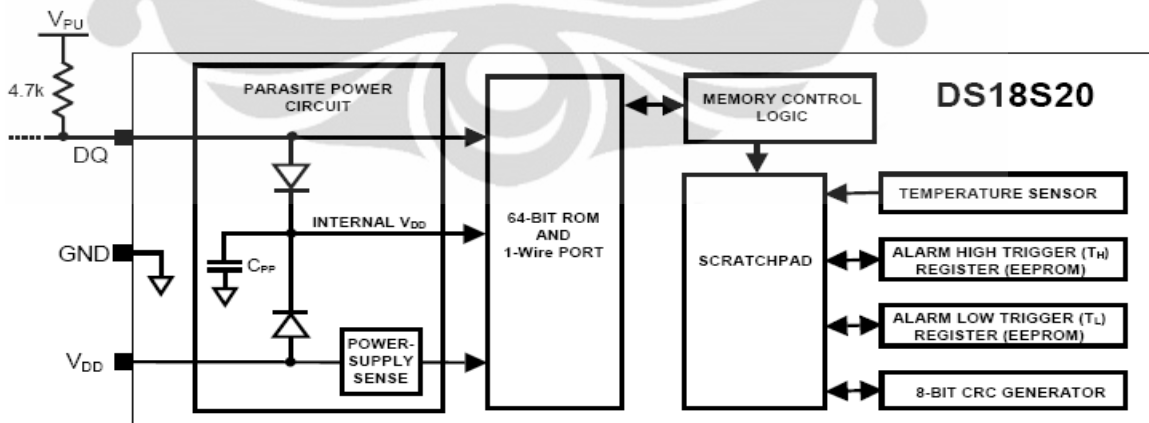


Gambar 2.2. Bentuk Fisik DS18B20

Adapun deskripsi pin dan blok diagram DS18B20 ditunjukkan oleh Tabel 2.1 dan Gambar 2.3.

Tabel 2. 1. Deskripsi pin DS18B20

Pin	Nama	Fungsi
1	GND	Ground
2	DQ	Data input/output
3	Vdd	Tegangan input

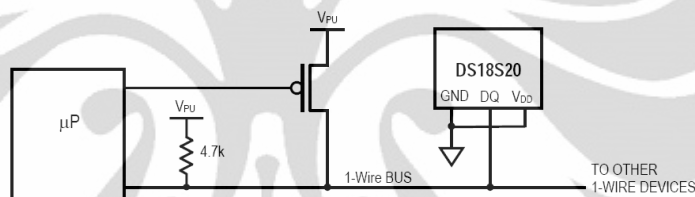


Gambar 2.3. Blok Diagram DS18B20

Pensuplaian pada DS18B20 terdapat 2 jenis *mode* yaitu pensuplaian dari luar (*external*) dan *mode* pensuplaian secara parasit (*parasite power*). Pada *mode*

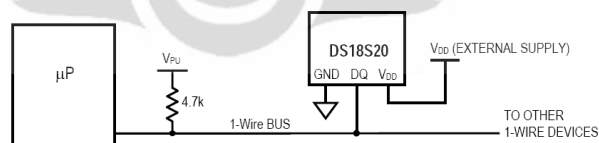
pen-suplaian dari luar maka suplai harus dihubungkan pada pin V_{dd} sedangkan jika menggunakan *mode parasit power* DS18B20 tidak memerlukan suplai dari luar [5].

Dengan menggunakan *mode parasit power* saat DS18B20 dalam proses pengkonversian temperatur atau menyalin data dari memori *scratchpad* ke EEPROM, arus yang beroperasi mencapai 1,5mA. Untuk memastikan bahwa DS18B20 mendapatkan arus yang cukup, maka diperlukan *pull-up* yang kuat pada jalur *1-wire*-nya, seperti yang ditunjukkan pada Gambar 2.4. Dengan menggunakan *mode parasite power* tidak direkomendasikan untuk pengukuran temperatur di atas 100⁰ C karena DS18B20 tidak mampu menahan komunikasi yang disebabkan kebocoran arus yang tinggi [5].



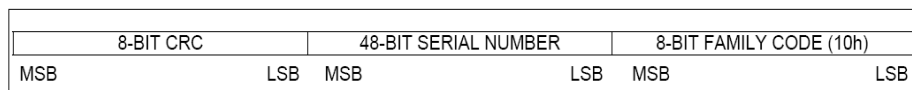
Gambar 2.4. Pensuplaian dengan *mode parasit power*

DS18B20 dapat juga diberikan *power* dengan metode konvensional dengan menghubungkan *power supply* luar dengan pin V_{dd} seperti yang ditunjukkan oleh Gambar 2.5. Untuk aplikasi pada temperatur tinggi tersebut sangat disarankan untuk menggunakan *power supply* dari luar [5].



Gambar 2.5. Pensuplaian dengan mode konvensional

Setiap DS18B20 memiliki 64 bit kode yang tersimpan di ROM dimana 8 bit pertama (LSB) merupakan kode *family* DS18B20 seperti yang terlihat pada Gambar 2.6. Pembacaan kode *family* ini ada pada perintah 28h, 48 bit selanjutnya merupakan nomor serial dan 8 bit terakhir (MSB) adalah *byte cyclic redundancy Check* (CRC) yang dihitung dari 56 bit pertama pada ROM *code* ini [5].

Gambar 2.6. 64-bit ROM *code* dari DS18B20

Semua transaksi DS18B20 pada jalur *1 wire* dimulai dengan langkah inisialisasi. Pada tahap inisialisasi dilakukan reset pulsa yang dikirim oleh *master* dan presense pulsa yang dikirim oleh *slave*. Setelah *master* mendeteksi pulsa yang muncul, maka akan berlanjut ke step ROM *command*. Perintah ini akan mengoperasikan kode 64 bit ROM pada setiap *device* dan *master* akan memilih *device* tertentu jika terdapat beberapa *device* dalam 1 jalur. Perintah ini juga memungkinkan *master* untuk menentukan jumlah dan *mode device* yang diperbolehkan dalam jalur *1-wire* [5].

Setelah jalur digunakan oleh ROM *command* untuk mengalamatkan DS18B20 agar dapat berkomunikasi, maka *master* dapat melakukan langkah *function command* pada DS18B20. Perintah-perintah ini memungkinkan *master* untuk menulis dan membaca data dari memori *scratchpad*, menginisialisasi pengkonversian temperatur, dan menentukan *mode power supply* [5].

2.3 Sensor Tekanan Udara Analog MPX4115A

Karakteristik dari sensor MPX4115A antara lain sebagai berikut :

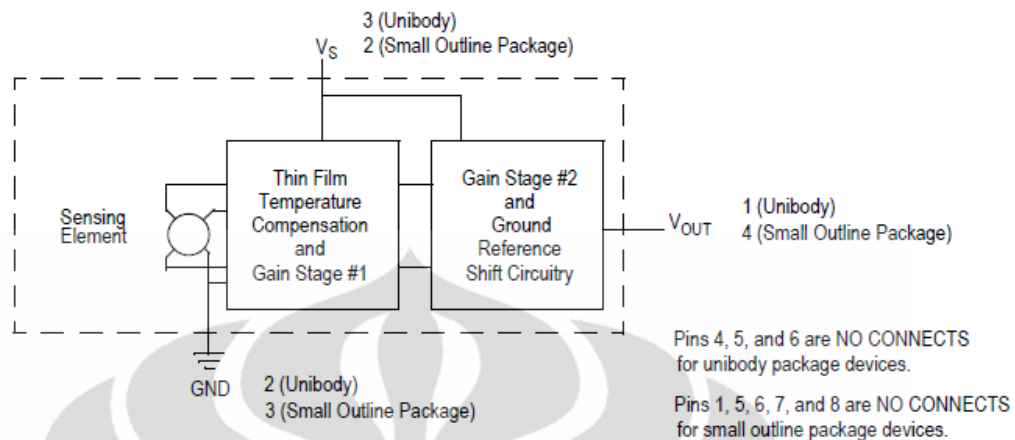
- a. Akurasi : $\pm 1,5 \% V_{FSS}$

Full Scale Span (VFSS) adalah *the algebraic difference* antara tegangan output disaat tekanan *full rated* dan tegangan output pada saat tekanan minimum (*minimum rated pressure*).

- b. Sensifitas : 46 mV/kPa

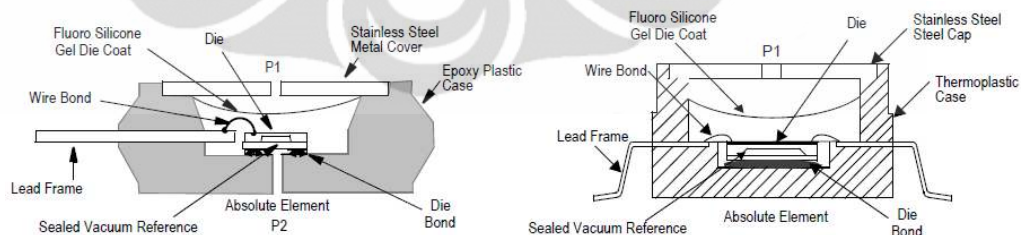
- c. Respon time : 1 ms

Sensor MPX4115A memiliki output analog sehingga harus diumpankan terlebih dahulu ke ADC mikrokontroler agar menjadi bentuk digital [6]. Gambar 2.7 menunjukkan blok diagram dari rangkaian internal yang terintegrasikan pada *chip* sensor tekanan.



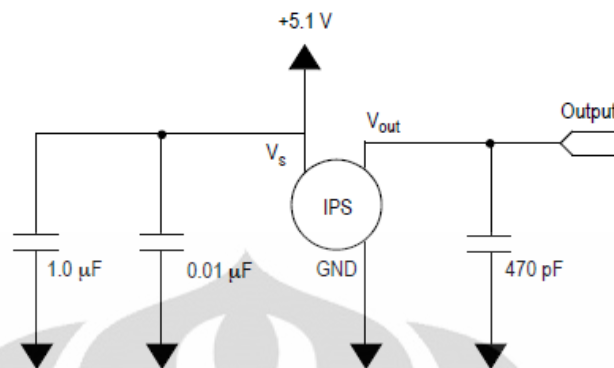
Gambar 2.7. Skematik sensor tekanan untuk *unibody package* dan *small outline package*

Sensor ini memiliki dua sisi, dimana sisi pertama merupakan sisi tekanan dan sisi kedua berupa vakum. Sensor dirancang untuk beroperasi dengan tegangan diferensial positif. Gambar 2.8 merupakan ilustrasi *absolute sensing chip* pada *basic chip carrier* (Case 867) dan *small outline chip carrier* (Case 482). Sebuah gel fluorosilicone mengisolasi permukaan mata dadu (*die surface*) dan *wire bonds* dari lingkungan, sementara itu sinyal udara yang diijinkan untuk dikirimkan ke diafragma sensor. Seri sensor tekanan udara MPX415A memiliki karakteristik operasi dan ketahanan internalnya dan tes kualifikasi berdasarkan penggunaan udara kering sebagai media tekanan. Media selain tekanan udara memungkinkan performa sensor menjadi kurang baik dan tahan lama [6].



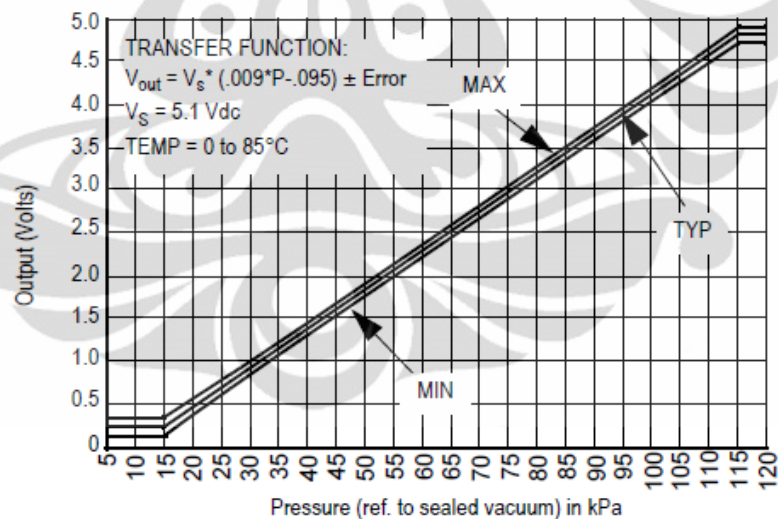
Gambar 2.8. *Cross-Sectional Diagram* MPX415A

Gambar 2.9 menunjukkan rangkaian decoupling yang direkomendasikan sebagai interfacing output dari sensor yang terintegrasi dengan input A/D dari mikrokontroler. Direkomendasikan decoupling dari suplai daya yang layak [6].



Gambar 2.9. Power supply decoupling dan filter output yang direkomendasikan untuk MPX4115A

Gambar 2.10 menunjukkan sinyal output relatif terhadap input tekanan. Secara khusus, kurva output minimum dan maksimum ditunjukkan untuk operasi temperatur 0°C hingga 85°C dengan menggunakan rangkaian decoupling seperti yang ditunjukkan pada Gambar 2.8. (Output akan mengalami saturasi diluar tekanan udara yang telah dispesifikasikan) [6].



Gambar 2.10. Sinyal output MPX4115A relatif terhadap tekanan udara

2.4 GPS

GPS adalah singkatan dari *Global Positioning System* yang merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tepat,

seluruhnya berjumlah 24 buah dimana 21 buah aktif bekerja dan 3 buah sisanya adalah cadangan [7].

2.4.1 Penentuan Posisi dengan GPS

Pada dasarnya penentuan posisi dengan GPS adalah pengukuran jarak secara bersama-sama ke beberapa satelit sekaligus. Untuk menentukan koordinat suatu titik di bumi, *receiver* setidaknya membutuhkan 4 satelit yang dapat ditangkap sinyalnya dengan baik. Secara *default*, posisi atau koordinat yang diperoleh bereferensi pada *global datum* yaitu *World Geodetic System 1984* [7].

2.4.2 Format Data GPS

Format data keluaran GPS ditetapkan oleh NMEA (*National Marine Electronic Association*). Format yang sering digunakan sebagai standar data keluaran GPS adalah NMEA 0813 yang berbentuk kalimat (*string*) yang merupakan rangkaian karakter ASCII 8 bit. Setiap kalimat diawali dengan satu karakter '\$', dua karakter *talker ID*, tiga karakter *sentence ID* dan diikuti oleh data *fields* yang masing-masing dipisahkan oleh koma serta diakhiri oleh *optional checksum* dan karakter *carriage return/line feed* (CR/LF). Jumlah maksimum karakter dihitung dari awal kalimat (\$) sampai dengan akhir kalimat (CR/LF) adalah 82 karakter [7]. Berikut ini adalah format data dasar NMEA 0813 :

$$\$aacc,c---c*hh<CR><LF>$$

Keterangan :

1. aa = *talker ID* menandakan jenis atau peralatan navigasi yang digunakan
2. ccc = *sentence ID* menandakan jenis informasi yang terkandung dalam kalimat
3. c---c = *data fields* yang berisi data-data navigasi hasil pengukuran
4. hh = *optional checksum* untuk pengecekan kesalahan (error) kalimat
5. <CR><LF> = menandakan akhir kalimat.

Jenis *talker ID* yang ada pada spesifikasi NMEA 0813 untuk data keluaran GPS *receiver* adalah GP. Sedangkan untuk jenis *sentence ID* terdapat tujuh macam data yang dapat ditampilkan [7], yaitu :

1. GGA adalah data tetap GPS
2. GLL adalah posisi geografis yaitu *latitude/longitude*
3. GSA adalah GNSS DOP dan satelit aktif yaitu penurunan akurasi dan jumlah satelit yang aktif pada *Global Satellite Navigation System*.
4. GSV adalah satelit GNSS dalam jangkauan.
5. RMC adalah spesifikasi data minimal GNSS yang direkomendasikan
6. VTG adalah jalur dan kecepatan
7. ZDA adalah waktu dan penanggalan.

2.4.3 GPS Garmin 18 LVC

Modul GPS yang digunakan dalam perancangan seismik data logger ini adalah Garmin 18 LVC. Interface pada tipe GPS ini dilakukan dengan menggunakan serial port dari komputer via konektor DB9 [8].



Gambar 2.11. GPS Garmin 18 LVC

Beberapa fitur GPS Garmin 18 LVC adalah :

- Memiliki berat dan panjang (kabel) yaitu 160 gram dan 5 meter
- Tegangan dan arus input sebesar 4 – 5,5 Vdc dan 90 mA @ 5 Vdc
- Update rate 1 record/sekon
- Akurasi receiver WAAS (Wide Area Argument System)
Posisi : < 3 meter, 95% typical dan Kecepatan : 0,1 knot RMS ready state
- Komunikasi protokol dengan NMEA (*National Marine Electronics Association*)

Tabel 2.2. NMEA *sentence* dalam bentuk \$GPGGA

\$GPGGA,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,.M,<10>.M,<11>,<12>*hh<CR><LF>

<1>	UTC time of position fix, hhmmss format for GPS 18 PC or LVC; hhmmss.s format for GPS 18-5Hz
<2>	Latitude, ddmn.mmmmm format for GPS 18 PC/LVC; ddmn.mmmmm for GPS 18-5Hz (leading zeros will be transmitted)
<3>	Latitude hemisphere, N or S
<4>	Longitude, dddmm.mmmmm format for GPS 18 PC/LVC; dddmm.mmmmm for GPS 18-5Hz (leading zeros will be transmitted)
<5>	Longitude hemisphere, E or W
<6>	GPS quality indication, 0 = fix not available, 1 = Non-differential GPS fix available, 2 = Differential GPS (WAAS) fix available, 6 = Estimated
<7>	Number of satellites in use, 00 to 12 (leading zeros will be transmitted)
<8>	Horizontal dilution of precision, 0.5 to 99.9
<9>	Antenna height above/below mean sea level, -9999.9 to 99999.9 meters
<10>	Geoidal height, -999.9 to 9999.9 meters
<11>	Null (Differential GPS)
<12>	Null (Differential Reference Station ID)

2.5 Mikrokontroler H8/3069F

Mikrokontroler adalah sebuah piranti elektronika digital yang yang terintegrasi dalam sebuah *chip* yang tersusun dari mikroprosesor dan piranti pendukungnya [9]. Ada perbedaan yang cukup penting antara mikroprosesor dan mikrokontroler. Jika mikroprosesor merupakan CPU (*Central Processing Unit*) tanpa memori dan I/O pendukung dari sebuah komputer sedangkan mikrokontroler umumnya terdiri dari CPU, memori, RAM, I/O tertentu dan unit pendukung, misalnya ADC (*Analog to Digital Converter*) yang sudah terintegrasi di dalam mikrokontroler tersebut.

Mikrokontroler yang digunakan pada penelitian ini adalah seri H8/300H yaitu *single-chip microcomputer* berperforma tinggi yang mempunyai sebuah CPU 16-bit (*Control Processing Unit*) sebagai prosessornya. Gambar 2.12 menunjukkan rangkaian minimum sistem mikrokontroler H8/3069F [10].

H8/3069F adalah salah satu mikrokontroler yang dilengkapi *processor* H8/300H. H8/3069F mempunyai sebuah *internal writable flash memory* yang menggunakan *single Power supply* (5V). Mikrokontroler ini dapat difungsikan sebagai *single-chip* mikrokomputer karena didalamnya sudah tersedia ROM, RAM dan fungsi I/O pada CPU. Mikrokontroler ini juga dapat difungsikan sebagai *multi-chip* mikrokomputer saat terjadi penambahan memori [10].



Gambar 2.12. Minimum sistem mikrokontroler H8/3069F

Berikut ini adalah beberapa fitur dari mikrokontroler H8/3069F :

2.5.1 *Internal ROM*

Mikrokontroler ini mempunyai flash memory 512K *byte* yang dapat di *Write* dengan sebuah *Power supply* (5V) [10].

2.5.2 *Internal RAM*

Mikrokontroler ini mempunyai 16K *byte internal RAM* [10].

2.5.3 *I/O port: 11 I/O port*

Tersedia 11 I/O port, yaitu port 1 sampai port 9, port A dan port B. Masing-masing port memiliki fungsi yang sama yaitu sebagai register, *port data register* (PDR) dan *port data direct register* (PDDR). Beberapa port juga bekerja sesuai *mode* yang ada dan beberapa port memiliki kelebihan sebagai input Pull-Up MOS Control Register (PCDR). I/O port dapat digunakan sebagai input status on/off atau sinyal dari berbagai sensor [10].

2.5.4 *Internal A/D converter dengan resolusi 10-bit × 8 channel*

8 *channel* analog input dibagi menjadi dua grup yaitu grup 0 dan grup 1. AV_{CC} dan AV_{SS} adalah *power supply* sirkuit analog pada *A/D converter*, dan V_{REF} adalah tegangan referensi [10].

2.5.5 Internal 8-bit D/A converter × 2 channel

Resolusi dari D/A *converter* ini adalah 8-bit. Tegangan output-nya berkisar antara 0V sampai V_{REF} . Pengaturan D/A *converter* ini diatur pada sebuah register di CPU [10].

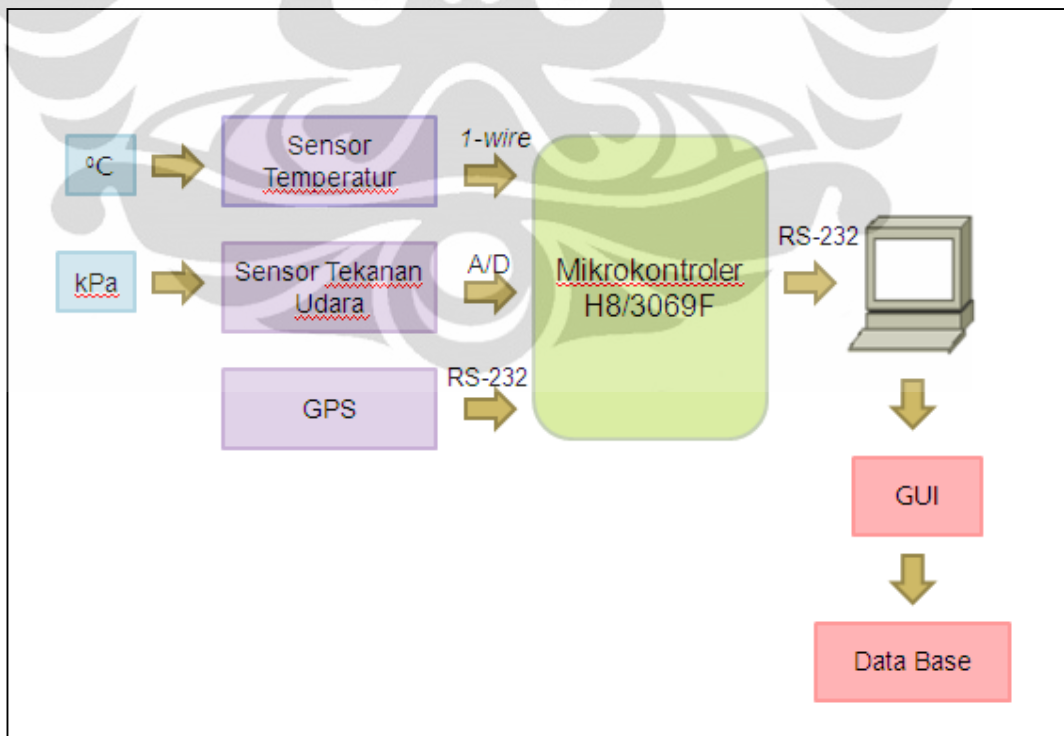


BAB 3 PERANCANGAN ALAT

3.1 PERANGKAT KERAS

Sistem akuisisi untuk monitoring temperatur dan tekanan udara menggunakan mikrokontroler H8/3069F yang kemudian diintegrasikan dengan sensor tekanan udara analog MPX4155A, sensor temperatur digital DS18B20, dan GPS Garmin 18 LVC.

Adapun koneksi mikrokontroler ke PC dihubungkan dengan menggunakan UART RS-232, seperti pada Gambar 3.1. Komunikasi pada RS-232 dengan PC adalah komunikasi asinkron [11]. Dimana sinyal *clock*nya tidak dikirim bersamaan dengan data. Format transmisi satu byte pada RS-232 adalah 8 bit, sebelum data tersebut ditransmisikan maka akan diawali oleh *start bit* dengan logika 0 (0 Volt), kemudian 8 bit data dan diakhiri oleh satu *stop bit* dengan logika 1 (5 Volt).



Gambar 3.1. Perancangan sistem akuisisi data

Adapun antar muka pengguna grafis menggunakan bahasa pemrograman Python. Python merupakan bahasa pemrograman yang berorientasi obyek dinamis, dapat digunakan untuk bermacam-macam pengembangan perangkat lunak. Selain itu, Python dapat berjalan di banyak *platform* [12].

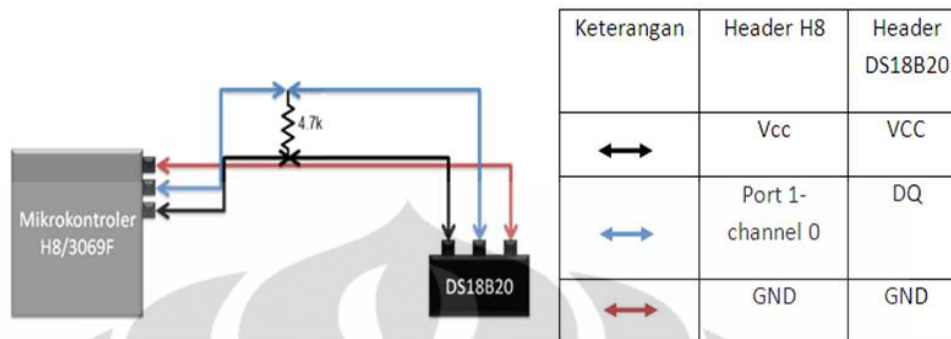
Sedangkan database untuk sistem akuisisi temperatur dan tekanan udara berbasis MySQL. Database server keluaran TcX, sebuah perusahaan IT Swedia ini menawarkan berbagai keunggulan dibandingkan database server lain. Sistem database MySQL memiliki sistem sekuritas dengan tiga verifikasi yaitu *username*, *password* dan *host*. Verifikasi *host* memungkinkan untuk membuka sekuriti di 'localhost', tetapi tertutup bagi *host* lain (bekerja di lokal komputer). Sistem pengaman ini ada di dalam database MySQL dan pada *tabel user* [13]. Berikut ini adalah beberapa keunggulan MySQL :

- a. MySQL merupakan program yang multi-threaded, sehingga dapat dipasang pada server yang memiliki multi-CPU.
- b. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL APIs dls.
- c. Bekerja pada berbagai platform. (tersedia berbagai versi untuk berbagai sistem operasi).
- d. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
- e. Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
- f. Mendukung ODBC untuk sistem operasi Microsoft Windows.
- g. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

3.1.1. Pengoperasian Sensor DS18B20

Pada penelitian ini pengoperasian DS18B20 menggunakan mode *power supply* dari luar sehingga pin Vcc pada DS18B20 dihubungkan ke Vcc mikrokontroler H8/3069F, pin DQ dihubungkan ke pin 0 pada port 1 mikrokontroler H8/3069F, dan pin *ground* DS18B20 dihubungkan ke *ground* mikrokontroler H8/3069F. Mode *power supply* dari luar sangat disarankan untuk untuk aplikasi pada temperatur tinggi diatas 100⁰C, seperti yang ditunjukkan oleh

Gambar 3.2.



Gambar 3.2 Rangkaian operasi sensor DS18B20 ke mikrokontroler H8/3069F dengan mode *power supply eksternal*

3.1.2 Konversi Analog ke Digital Menggunakan Mikrokontroler H8/3069F

Sebelum diproses oleh sebuah *processor* (mikrokontroler), pada umumnya masukan sinyal analog yang telah melewati suatu pengkondisi sinyal dikonversi terlebih dahulu menjadi data digital. Konversi ini dilakukan oleh ADC (*Analog to Digital Converter*). Analog to Digital Converter adalah suatu rangkaian yang berfungsi mengubah sinyal analog menjadi sinyal digital. Pada penelitian ini digunakan ADC internal mikrokontroler H8/3069F. ADC ini adalah ADC 10 bit yang menggunakan metode *successive-approximations*. Mikrokontroler H8/3069F menyediakan delapan buah kanal input konversi.

ADC internal mikrokontroler H8/3069F memiliki empat buah data register yaitu ADDRA sampai ADDRD yang masing-masing merupakan 16 bit *read only register* yang menyimpan hasil konversi. Setiap konversi akan menghasilkan data 10 bit yang kemudian akan disimpan ke dalam data register bergantung kanal yang dipilih. ADC H8/3069F menyediakan delapan pin kanal input yang dibagi menjadi dua grup, yaitu :

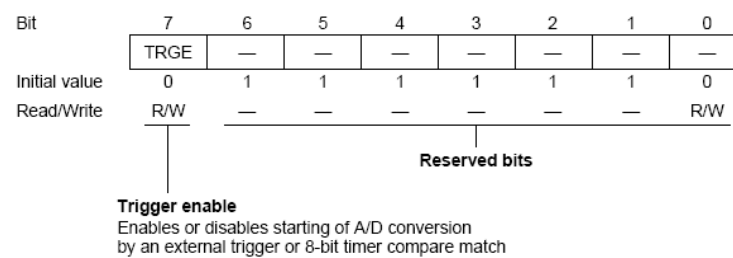
- Grup 0 : terdiri atas AN0-AN3, dan
- Grup 1 : terdiri atas AN4-AN7.

Tabel 3.1 Konfigurasi pin ADC internal mikrokontroler H8/3069F

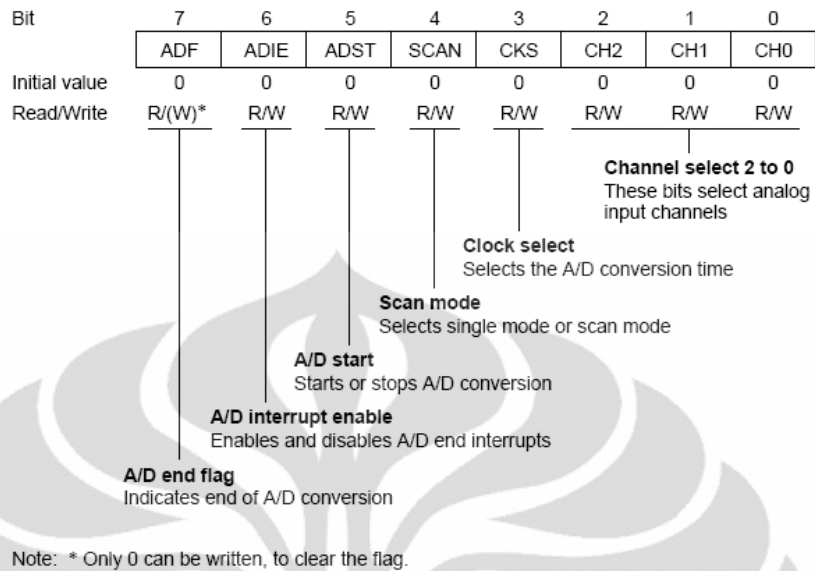
Pin	Simbol	I/O	Fungsi

Power suplai	AVcc	Input	Power suplai analog
Analog ground	Avss	Input	Analog ground
Tegangan referensi	Vref	Input	Tegangan referensi
Input analog 0	AN0	Input	Input analog grup 0
Input analog 1	AN1	Input	
Input analog 2	AN2	Input	
Input analog 3	AN3	Input	
Input analog 4	AN4	Input	Input analog grup 1
Input analog 5	AN5	Input	
Input analog 6	AN6	Input	
Input analog 7	AN7	Input	
Eksternal trigger	ADGRT	Input	Pemicu eksternal konversi

Dalam fungsi control ADC, mikrokontroler H8/3069F memiliki dua buah register, yaitu ADCR (*Analog to Digital Control Register*) dan ADCSR (*Analog to Digital Control/Status Register*). ADCSR adalah register *readable/writable* 8 bit yang berfungsi untuk memilih mode dan mengontrol konversi sedangkan ADCR adalah register *readable/writable* 8 bit yang berfungsi untuk mengaktifkan eksternal trigger atau membandingkan konversi dengan timer 8 bit (Gambar 3.3 dan 3.4).



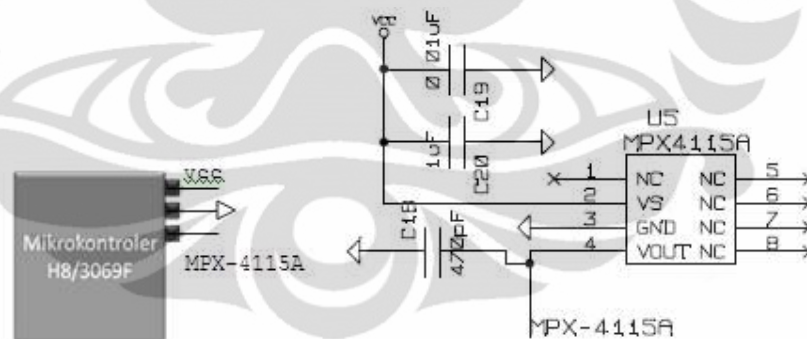
Gambar 3.3. A/D control register (ADCR) H8/3069F



Gambar 3.4. A/D control/status register (ADCSR) H8/3069F

3.1.3 Pengoperasian Sensor MPX4115A

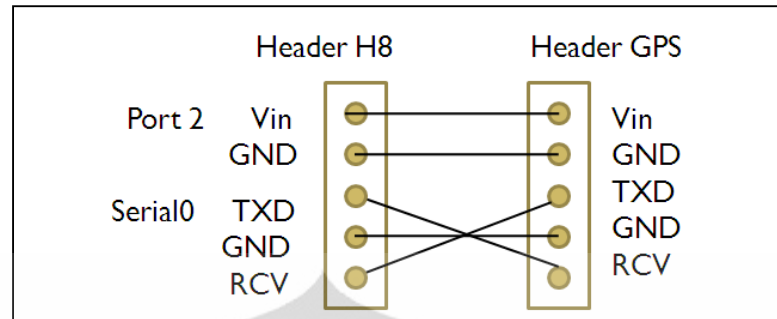
Sensor MPX4115A membutuhkan rangkaian pengkondisi sinyal sebelum dikoneksikan ke modul mikrokontroler H8/3069 F, seperti yang ditunjukkan pada Gambar 3.5.



Gambar 3.5. Rangkaian operasi sensor MPX4115A ke mikrokontroler H8/3069F

3.1.4 Pengoperasian GPS Garmin 18 LVC

GPS Garmin 18 LVC dapat dikoneksikan langsung ke mikrokontroler dengan menghubungkan RX, TX dan GND GPS ke RX, TX dan GND mikrokontroler H8/3069F. Untuk mendapatkan waktu *real time* GPS yang nantinya akan di olah oleh mikrokontroler untuk ditampilkan oleh PC diperlukan adanya sentence NMEA atau pernyataan dari GPS tersebut, yaitu \$GPGGA.

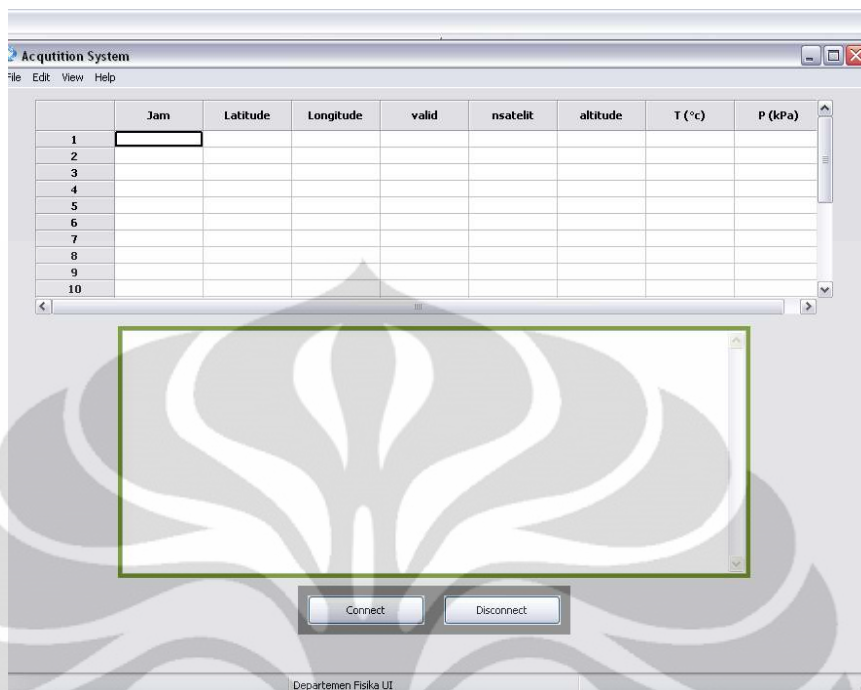


Gambar 3.6. Interkoneksi GPS Garmin 18 LVC ke Mikrokontroler H8/3069F

3.2 Perangkat Lunak

Perangkat lunak terdiri dari pemrograman dengan bahasa C untuk mengoperasikan sensor temperatur digital DS18B20 dengan komunikasi ke mikrokontroler H8/3069F dengan interface *1-wire*, ADC mikrokontroler H8/3069F untuk mengoperasikan sensor tekanan udara MPX4115A. Pemrograman ADC internal mikrokontroler H8/3069F yang dirancang adalah *scan mode*. Selain itu, pemrograman untuk *parsing* data GPS untuk mengambil data waktu, tanggal dan posisi saja. Kesemuanya diuji satu per satu, kemudian diintegrasikan.

Selanjutnya, data tersebut akan dikirim ke PC melalui koneksi serial dan ditampilkan pada GUI dengan menggunakan bahasa pemrograman Python yang dikategorikan sebagai bahasa tingkat tinggi. *Library* Python yang digunakan untuk membuat tampilan GUI pada penelitian ini adalah wxPython. Gambar 3.7 merupakan tampilan GUI sebelum komputer dihubungkan dengan mikrokontroler H8/3069F melalui serial.



Gambar 3.7. Tampilan GUI sebelum di hubungkan ke mikrokontroler berbasis Python

3.3 Pengujian Sensor Temperatur Digital DS18B20

Pengoperasian DS18B20 hanya membutuhkan 1 pin I/O mikrokontroler H8/3069F sebagai jalur data input dan output. Untuk mengaktifkan komunikasi DS18B20 secara *1 wire* dengan mikrokontroler H8/3069F maka peneliti mengisikan program dengan bahasa pemrograman C.

Berikut ini merupakan langkah-langkah untuk mengisikan sebuah program pada mikrokontroler H8/3069F agar DS18B20 dapat beroperasi, sebelum melakukan *writing* hasil kompilasi ke modul mikrokontroler H8/3069F, peneliti mengatur posisi DIP *switch* ke mode-3 seperti pada tabel 3.2 sebagai berikut :

Tabel 3.2. Posisi DIP Switch Mode-3

	Pin 1	Pin 2	Pin 3	Pin 4
ON	O	O		O
OFF			O	

Langkah selanjutnya adalah mengisikan sebuah program pada mikrokontroler H8/3069F agar kode *family* atau ROM Code dari DS18B20 dapat terbaca.

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\ani>d:
D:\>cd h8
D:\h8>cd 1wire
D:\h8\1wire>ls
Makefile  ds1820.c      interrupt.h  main.map    onewire.c   start.S     vects.c
delay.c   ds1820.h      ioh83069.h main.mot    onewire.h   uart.c
delay.h   h8write.exe  main.c      main.out    rom3068f.ld uart.h

D:\h8\1wire>h8write.exe -3069 -f20 main.mot com3
H8/3069F is ready! 2002/5/20 Yukio Mituiwa.
writing
WARNING:This Line dosen't start with"S".
Address Size seems wrong
WARNING:This Line dosen't start with"S".
Address Size seems wrong
.....
EEPROM Writing is succeeded.
D:\h8\1wire>

```

Gambar 3.8. Langkah-langkah saat melakukan *writing* program pembacaan ROM Code DS18B20 ke mikrokontroler H8/3069F

Setelah melakukan *writing* ubahlah posisi DIP *switch* ke mode-7 untuk melakukan *reading* seperti yang ditunjukkan pada Tabel 3.3 [14]

Tabel 3.3. Posisi DIP Switch Mode-7

	Pin 1	Pin 2	Pin 3	Pin 4
ON	O	O	O	
OFF				O

Pengisian program ke modul mikrokontroler H8/3069F untuk akuisisi temperatur setelah diketahui ROM Code dari DS18B20 dengan mode DS18B20 sudah 9 bit.


```

C:\> Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\ami>d:
D:\>cd h8
D:\h8>cd 1wire
D:\h8\1wire>ls
Makefile  ds1820.c      interrupt.h  main.map    onewire.c   start.S     vects.c
delay.c   ds1820.h      ioh83069.h main.mot    onewire.h   uart.c
delay.h   h8write.exe  main.c      main.out    rom3068f.ld uart.h

D:\h8\1wire>h8write.exe -3069 -f20 main.mot com3
H8/3069F is ready! 2002/5/20 Yukio Mituiwa.
writing
WARNING:This Line dosen't start with"S".
Address $ize seems wrong
WARNING:This Line dosen't start with"S".
Address $ize seems wrong
.....
EEPROM Writing is succeeded.
D:\h8\1wire>

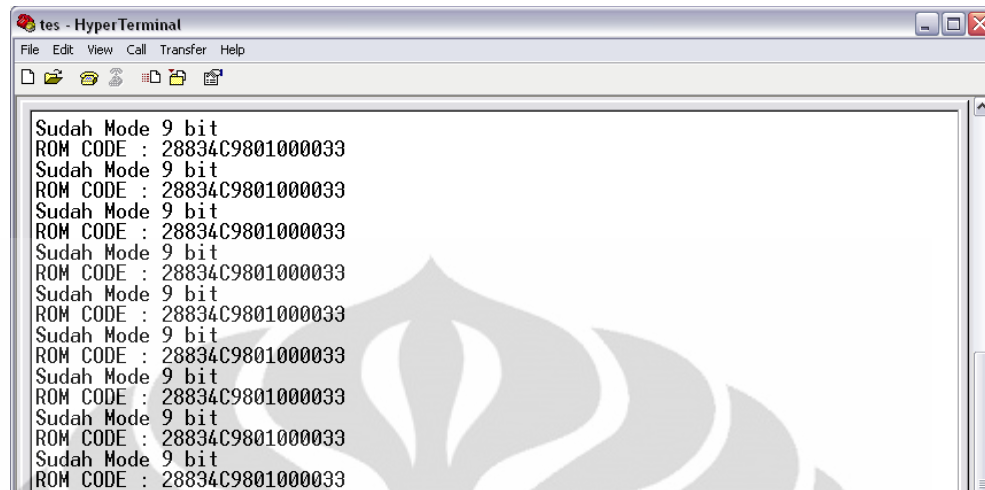
```

Gambar 3.9. Langkah-langkah saat melakukan *writing* program pengoperasian DS18B20 ke mikrokontroler H8/3069F

3.3.1 Hasil dan Analisis Data Rom Code DS18B20

DS18B20 (sensor temperatur digital) menggunakan komunikasi *1 wire* untuk *interfacing* dengan mikrokontroler. Komunikasi *1 wire* dirancang untuk transfer data yang menggabungkan sinyal data dan *power* dalam satu jalur. Kelebihannya terletak pada jalur data yang digunakan yakni hanya satu jalur *power*, data, dan satu jalur *ground*. Model komunikasi *1 wire* ini adalah *master-slave*, dimana mikrokontroler H8/3069F merupakan masternya dan DS18B20 adalah *device 1 wire* yang menjadi *slavenya*.

DS18B20 memiliki 64 bit kode unik yang tersimpan di ROM. 8 bit pertama (LSB) merupakan kode *family* DS18B20 seperti yang terlihat pada Gambar 3.11. Pembacaan kode *family* ini ada pada perintah 28h, 48 bit selanjutnya merupakan nomor serial dan 8 bit terakhir (MSB) adalah *byte cyclic redundancy Check* (CRC) yang dihitung dari 56 bit pertama pada ROM *code* ini. Selain itu, pada Gambar terlihat tulisan “*Sudah Mode 9 bit*”. Hal ini berarti resolusi ADC internal DS18B20 dirubah ke dalam mode 9 bit .



```

tes - HyperTerminal
File Edit View Call Transfer Help
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033
Sudah Mode 9 bit
ROM CODE : 28834C980100033

```

Gambar 3.10. *Family code* ROM DS18B20

3.3.2 Hasil dan Analisis Data Temperatur

Setelah mendapatkan kode *family* dari sensor DS18B20 maka pemrograman akuisisi temperatur diisikan ke mikrokontroler H8/3069F. Data awal yang diperoleh adalah dalam bentuk hexadecimal lalu dirubah ke dalam biner. Semua transaksi pada jalur *1 wire* dimulai dengan langkah inisialisasi. Pada tahap inisialisasi dilakukan reset pulsa yang dikirim oleh *master* dan presense pulsa yang dikirim oleh *slave*. Setelah *master* mendeteksi pulsa yang muncul, maka akan berlanjut ke step ROM *command*. Perintah ini akan mengoperasikan kode 64 bit ROM pada setiap *device* dan *master* akan memilih *device* tertentu jika terdapat beberapa *device* dalam 1 jalur.

Pada saat DS18B20 beroperasi maka akan terjadi proses pengkonversian temperatur dan konversi ADC pada perintah 44h. Perintah ini merupakan penginisialisasian untuk proses pengkonversian temperatur. Hasil data temperatur disimpan dalam 2 *byte* register temperatur dalam memori *scratchpad*. DS18B20 akan merespon dengan mengirim bit 0 saat pengkonversian temperatur masih dalam proses dan mengirimkan bit 1 saat pengkonversian telah selesai.

Gambar 3.12 menunjukkan potongan data yang ditampilkan di hyperterminal. Berdasarkan data temperatur yang terbaca di hyperterminal, maka dapat dipastikan bahwa program DS18B20 yang telah dibuat sudah dapat dipakai untuk membaca temperatur.



```

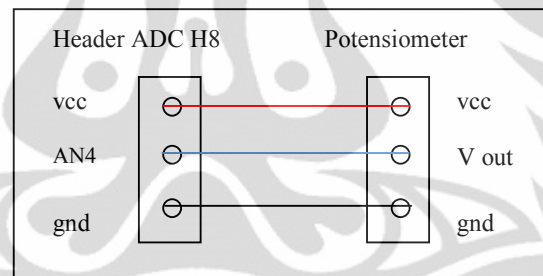
tes - HyperTerminal
File Edit View Call Transfer Help
Inialisasi Uart
Inialisasi Ds18B20
27.00
27.00
27.00
27.00
27.00
27.00
27.00
27.00
27.00
27.00
27.00
27.00

```

Gambar 3.11. Pengukuran suhu kamar dengan menggunakan DS18B20

3.4 Uji linearitas dari rangkaian ADC (*Analog to Digital Converter*) internal mikrokontroler H8/3069F

Pengujian ADC internal mikrokontroler H8/3069F dilakukan dengan menggunakan potensiometer 10 k Ω yang dihubungkan ke pin mikrokontroler H8/3069F.



Gambar 3.12. Interkoneksi ADC mikrokontroler H8/3069F dengan potensiometer

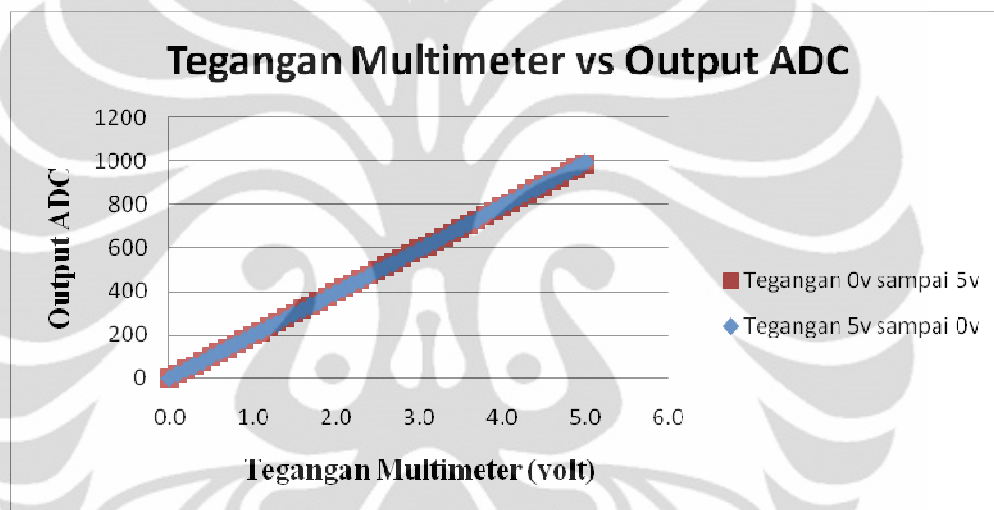
Potensiometer diatur untuk mendapatkan variasi tegangan output dari 0 volt hingga 5 volt. Kemudian potensiometer diatur kembali untuk mendapatkan tegangan output maksimum (5 volt) hingga tegangan output minimum (0 volt). Adapun tegangan referensi ADC adalah 5.2 volt. Berikut ini adalah tabel hasil konversi ADC sesuai perubahan tegangan yang diberikan :

Tabel 3.4. (a) Konversi ADC dengan tegangan output dari 0 V sampai 5 V (b) Konversi ADC dengan tegangan output dari 5 V sampai 0 V

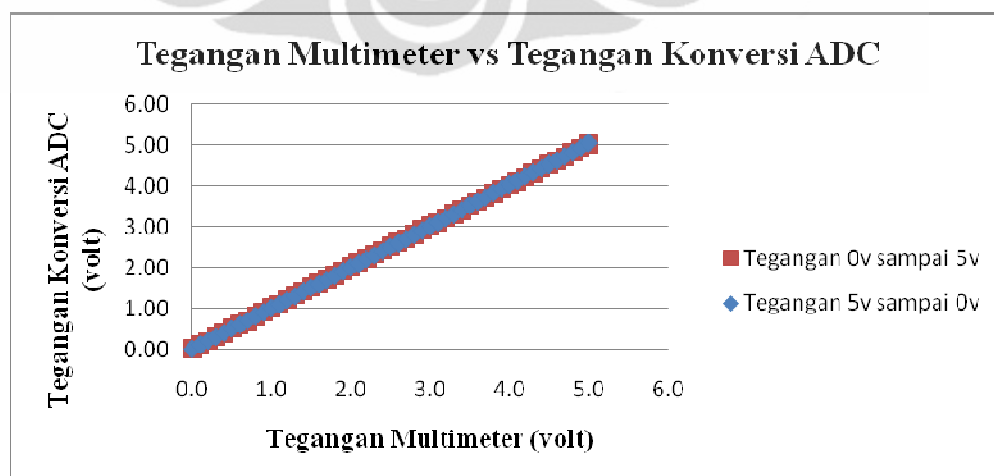
(a)			(b)		
Tegangan Output (volt)	Hasil konversi ADC	Tegangan Input ADC (volt)	Tegangan Output (volt)	Hasil konversi ADC	Tegangan Input ADC (volt)
0.0	2	0.01	0.0	2	0.01
0.1	22	0.11	0.1	22	0.11
0.2	41	0.21	0.2	41	0.21
0.3	61	0.31	0.3	60	0.30
0.4	80	0.41	0.4	79	0.40
0.5	102	0.52	0.5	100	0.51
0.6	119	0.60	0.6	119	0.60
0.7	137	0.70	0.7	139	0.71
0.8	161	0.82	0.8	160	0.81
0.9	179	0.91	0.9	178	0.90
1.0	200	1.02	1.0	198	1.01
1.1	215	1.09	1.1	216	1.10
1.2	241	1.22	1.2	237	1.20
1.3	260	1.32	1.3	255	1.29
1.4	279	1.42	1.4	278	1.41
1.5	299	1.52	1.5	296	1.50
1.6	319	1.62	1.6	314	1.59
1.7	338	1.72	1.7	334	1.70
1.8	357	1.81	1.8	354	1.80
1.9	377	1.91	1.9	375	1.90
2.0	398	2.02	2.0	393	2.00
2.1	417	2.12	2.1	412	2.09
2.2	437	2.22	2.2	433	2.20
2.3	458	2.33	2.3	452	2.30
2.4	477	2.42	2.4	471	2.39
2.5	497	2.52	2.5	492	2.50
2.6	518	2.63	2.6	511	2.59
2.7	534	2.71	2.7	530	2.69
2.8	555	2.82	2.8	551	2.80
2.9	576	2.93	2.9	570	2.89
3.0	596	3.03	3.0	590	3.00
3.1	613	3.11	3.1	609	3.09
3.2	632	3.21	3.2	630	3.20
3.3	652	3.31	3.3	649	3.30
3.4	670	3.40	3.4	669	3.40
3.5	690	3.50	3.5	691	3.51
3.6	712	3.62	3.6	711	3.61
3.7	732	3.72	3.7	730	3.71
3.8	750	3.81	3.8	752	3.82
3.9	771	3.92	3.9	771	3.92
4.0	790	4.01	4.0	792	4.02

4.1	810	4.11
4.2	830	4.21
4.3	850	4.32
4.4	871	4.42
4.5	889	4.51
4.6	910	4.62
4.7	929	4.72
4.8	952	4.83
4.9	968	4.92
5.0	990	5.03

4.1	811	4.12
4.2	831	4.22
4.3	850	4.32
4.4	868	4.41
4.5	889	4.51
4.6	908	4.61
4.7	928	4.71
4.8	946	4.80
4.9	968	4.92
5.0	992	5.04



Gambar 3.13. Grafik perbandingan output potensiometer pada multimeter dan output ADC



Gambar 3.14. Grafik perbandingan output potensiometer pada multimeter dan hasil konversi

Grafik diatas menunjukkan bahwa output yang dihasilkan cukup linear dengan input yang diberikan. ADC internal mikrokontroler H8/3069F memiliki resolusi 10 bit. Hal ini berarti ADC tersebut dapat menghasilkan output 10 kode digital. Nilai ADC maksimum dari mikrokontroler H8/3069F adalah 1023. Hal ini berarti ADC ini memiliki sensitivitas $5.2/1023 \approx 0.005$ V. Adapun tegangan output yang keluar dari sensor akan menjadi tegangan input untuk ADC H8/3069F. Berikut ini adalah persamaan matematis untuk mengetahui tegangan input ADC.

$$\text{Tegangan input} = \frac{\text{nilai ADC}}{2^n} \times \text{tegangan referensi ADC} \quad (4.1)$$

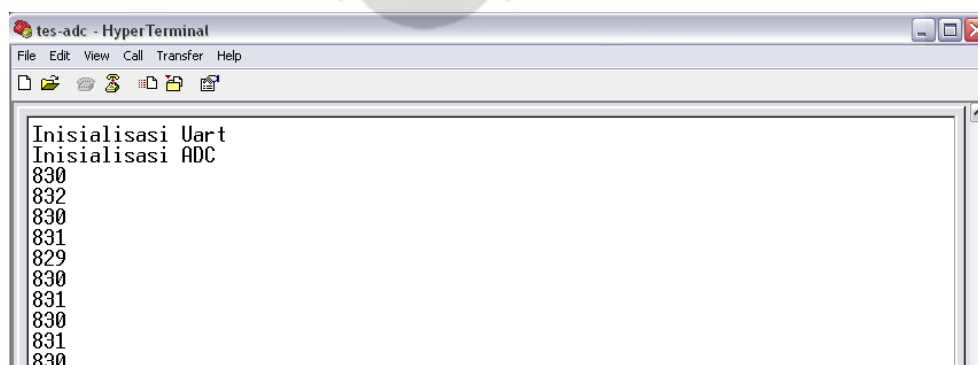
$$\text{Tegangan input} = \frac{\text{nilai ADC}}{2^{10}} \times 5.2 \text{ volt} \quad (4.2)$$

3.5 Hasil dan Analisis Tekanan Udara

Akuisisi tekanan udara pada penelitian kali ini menggunakan sensor tekanan udara analog MPX4115A. Sensor ini membutuhkan *power supply* sebesar 5 volt dan pemrograman pada ADC internal untuk mengubah data analog dari sensor tersebut menjadi data digital. Berikut ini adalah hasil pengujian output ADC yang terukur dari sensor MPX4115A. Berikut ini adalah fungsi transfer yang digunakan untuk mengubah data ADC menjadi satuan kPa.

$$\text{Pressure} = ((V_{out}/V_{in}) + 0.095) / 0.009 \quad (4.3)$$

$$\text{Pressure} = ((\text{analogRead}/1023) + 0.095) / 0.009 \quad (4.4)$$



```
tes-adc - HyperTerminal
File Edit View Call Transfer Help
Inisialisasi Uart
Inisialisasi ADC
830
832
830
831
829
830
831
830
831
830
```

Gambar 3.15. Nilai tekanan udara dari sensor MPX4115A

3.6 Hasil dan Analisis Pengoperasian GPS Garmin 18 LVC

GPS Garmin 18LVC dapat menampilkan format NMEA dalam bentuk format GPGSV, GPRMC, GPGSA, dan GPGGA. Karena hanya salah satu format saja yang akan ditampilkan dan diproses oleh mikrokontroler H8/3069F, maka dilakukan set up pada GPS Garmin 18LVC. Karena data yang diperlukan dari GPS adalah berupa data waktu, posisi, dan ketinggian. Maka dilakukan perubahan *set up* pada GPS GARMIN 18LVC agar NMEA *sentences* yang ditampilkan dalam format GPGGA saja.

```

tes - HyperTerminal
File Edit View Call Transfer Help
$GPGGA,083336,0622.1806,S,10649.6700,E,1,09,1.1,96.7,M,7.3,M,,*55
$GPGGA,083337,0622.1806,S,10649.6700,E,1,09,1.1,96.7,M,7.3,M,,*54
$GPGGA,083338,0622.1805,S,10649.6701,E,1,09,1.0,96.7,M,7.3,M,,*58
$GPGGA,083339,0622.1805,S,10649.6701,E,1,09,1.1,96.7,M,7.3,M,,*58
$GPGGA,083340,0622.1805,S,10649.6701,E,1,09,1.0,96.6,M,7.3,M,,*56
$GPGGA,083341,0622.1805,S,10649.6701,E,1,09,1.0,96.6,M,7.3,M,,*57
$GPGGA,083342,0622.1805,S,10649.6702,E,1,09,1.0,96.6,M,7.3,M,,*57
$GPGGA,083343,0622.1805,S,10649.6702,E,1,09,1.0,96.6,M,7.3,M,,*56
$GPGGA,083344,0622.1805,S,10649.6702,E,1,09,1.0,96.6,M,7.3,M,,*51
$GPGGA,083345,0622.1805,S,10649.6702,E,1,09,1.0,96.6,M,7.3,M,,*50
$GPGGA,083346,0622.1805,S,10649.6702,E,1,08,1.3,96.6,M,7.3,M,,*51
$GPGGA,083347,0622.1805,S,10649.6703,E,1,09,1.0,96.6,M,7.3,M,,*53
$GPGGA,083348,0622.1804,S,10649.6703,E,1,09,1.0,96.6,M,7.3,M,,*5D
$GPGGA,083349,0622.1804,S,10649.6703,E,1,09,1.0,96.6,M,7.3,M,,*5C
$GPGGA,083350,0622.1804,S,10649.6703,E,1,09,1.0,96.5,M,7.3,M,,*57
$GPGGA,083351,0622.1804,S,10649.6704,E,1,09,1.0,96.5,M,7.3,M,,*51
$GPGGA,083352,0622.1804,S,10649.6704,E,1,09,1.2,96.5,M,7.3,M,,*50
$GPGGA,083353,0622.1803,S,10649.6704,E,1,09,1.0,96.5,M,7.3,M,,*54
$GPGGA,083354,0622.1803,S,10649.6704,E,1,09,1.0,96.5,M,7.3,M,,*53
$GPGGA,083355,0622.1803,S,10649.6705,E,1,09,1.2,96.5,M,7.3,M,,*51
$GPGGA,083356,0622.1803,S,10649.6705,E,1,09,1.1,96.5,M,7.3,M,,*51
$GPGGA,083357,0622.1803,S,10649.6705,E,1,09,1.2,96.4,M,7.3,M,,*52
$GPGGA,083358,0622.1803,S,10649.6705,E,1,09,1.2,96.4,M,7.3,M,,*5D
Connected 0:27:37 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

```

Gambar 3.16. Hasil porting GPS Garmin 18LVC ke mikrokontroler H8/3069F untuk format GPGGA

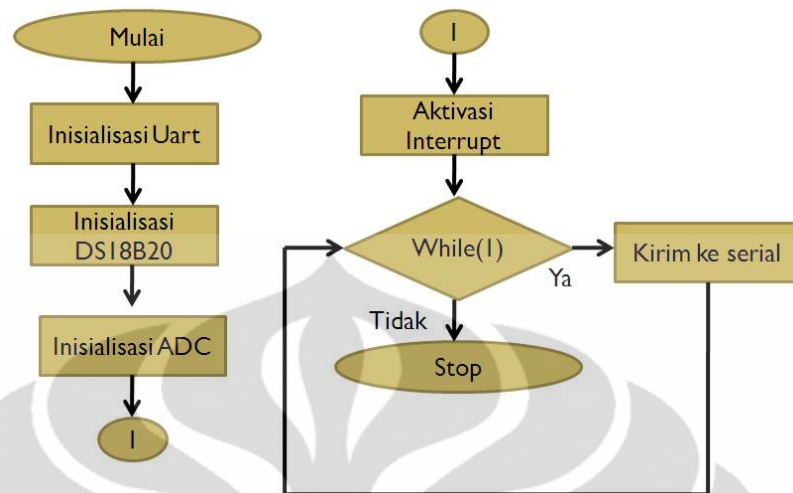
Tabel 3.5. Format NMEA 0813 dalam bentuk GPGGA

\$GPGGA,083358,0622.1803,S,10649.6705,E,1,09,1.2,96.4,M,7.3,M,,*5D			
Nama	Contoh	Satuan	Keterangan
Jenis Kalimat	\$GPGGA		GGA Protocol Header
Waktu UTC	083358		Hhmmss
Garis Lintang	0621.6240		Ddmm,mmmm

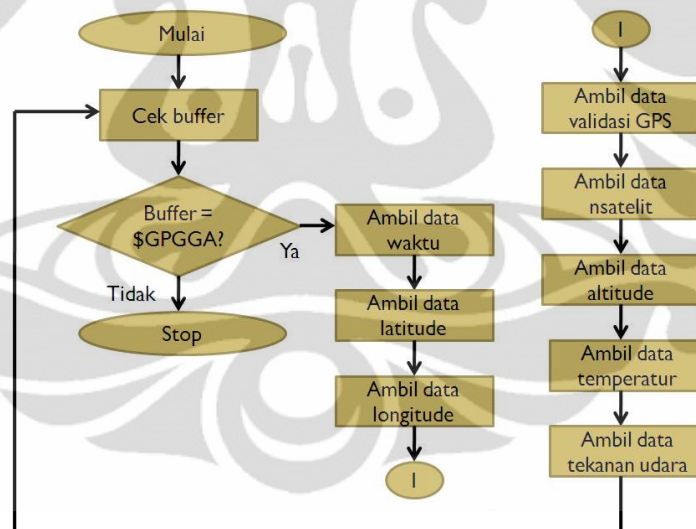
N/S	S		N = utara / S =selatan
Garis Bujur	10643,5598		Dddmm,mmmm
E/W	E		E = timur / W = barat
Status indikator kualitas GPS	0 1 2		GPS belum fix GPS sudah fix GPS diferensial fix
Jumlah Satelit	09		00-12 satelit yang dapat digunakan
Presisi dari <i>horizontal dilution (HDOP)</i>	2		
Altitude	96,4	meter	
Unit	M	meter	
<i>Geoidal separation</i>	7,3	meter	
Unit	M	meter	
<i>Differential GPS</i>			Field kosong berarti DGPS tidak digunakan
<i>Diferential reference station</i>			0000-1023
<i>Checksum</i> <CR><LF>	*5D		Akhir Kalimat

3.7 Hasil dan Analisis Integrasi Sensor-Sensor dengan GPS

Setelah proses pengujian sensor dan GPS dilakukan, maka langkah selanjutnya dilakukan *parsing* data GPS agar data yang tampil hanya waktu, tanggal dan posisi garis lintang dan garis bujur. Kemudian data GPS yang telah di-*parsing*, digabung dengan sensor DS18B20 dan MPX4115A. Berikut ini adalah *flowchart* untuk akuisisi data temperatur dan tekanan udara.



Gambar 3.17. Diagram alir program utama

Gambar 3.18. Diagram alir program untuk aktivasi *interrupt*

Berikut ini adalah langkah-langkah saat melakukan *writing* program integrasi sensor-sensor dan GPS hasil pengujian dalam skala laboratorium, dimana komunikasi *hardware* ke PC menggunakan RS-232.

```

Command Prompt
C:\Documents and Settings\ami>d:
D:\>cd h8
D:\h8>cd ami
D:\h8\ami>ls
Makefile  delay.h      interrupt.h  main.mot    rom3068f.ld  uart.h
adc.c     ds1820.c    ioh83069.h  main.out    romcode.h    vects.c
adc.h     ds1820.h    main.c      onewire.c   start.S
delay.c   h8write.exe main.map    onewire.h   uart.c

D:\h8\ami>h8write.exe -3069 -f20 main.mot com3
H8/3069F is ready! 2002/5/20 Yukio Mituiwa.
writing
WARNING:This Line dosen't start with"S".
Address Size seems wrong
WARNING:This Line dosen't start with"S".
Address Size seems wrong
.....
EEPROM Writing is succeeded.
D:\h8\ami>

```

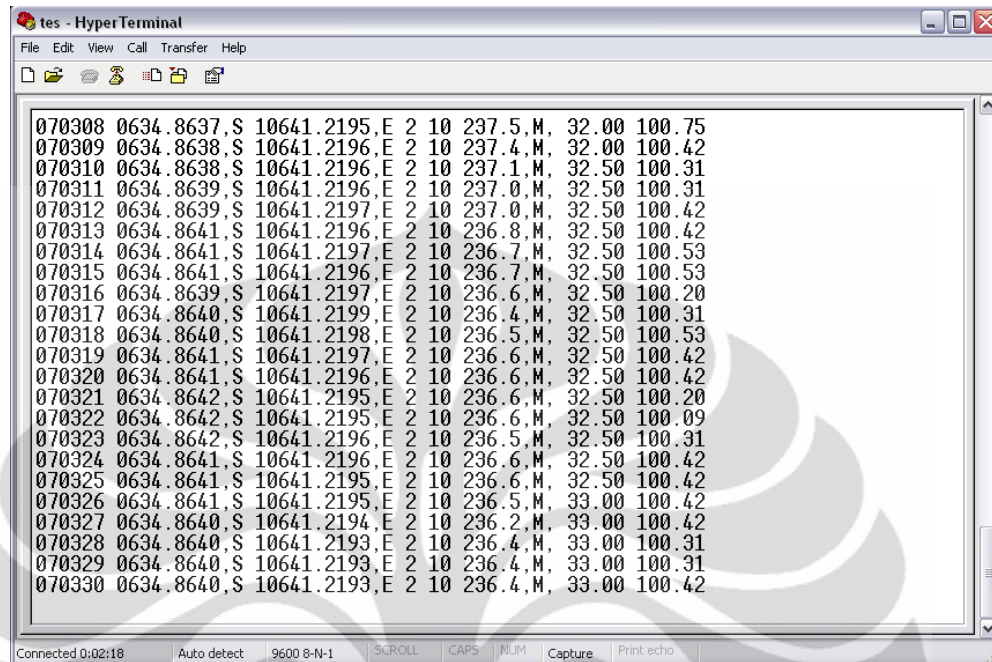
Gambar 3.19. Langkah-langkah saat melakukan *writing* program integrasi sensor-sensor dan GPS

Gambar 3.20 menunjukkan hasil pengujian perangkat keras yang sudah terintegrasi. Berikut ini adalah tampilan interkoneksi sistem perangkat keras di lapangan.



Gambar 3.20. Interkoneksi awal sistem perangkat keras (1) Mikrokontroler H8/3069F (2) Sensor DS18B20 (3) GPA Garmin 18 LVC (4) Aki 5Ah/12V (5) Sensor MPX4155A (6) Rangkaian Regulator

Berturut-turut kolom tersebut menyatakan jam, garis lintang (latitude), garis bujur (longitude), validasi GPS, jumlah satelit yang digunakan, altitude (m), data temperatur ($^{\circ}\text{C}$) dan tekanan udara (kPa).



```
tes - HyperTerminal
File Edit View Call Transfer Help
070308 0634.8637,S 10641.2195,E 2 10 237.5,M, 32.00 100.75
070309 0634.8638,S 10641.2196,E 2 10 237.4,M, 32.00 100.42
070310 0634.8638,S 10641.2196,E 2 10 237.1,M, 32.50 100.31
070311 0634.8639,S 10641.2196,E 2 10 237.0,M, 32.50 100.31
070312 0634.8639,S 10641.2197,E 2 10 237.0,M, 32.50 100.42
070313 0634.8641,S 10641.2196,E 2 10 236.8,M, 32.50 100.42
070314 0634.8641,S 10641.2197,E 2 10 236.7,M, 32.50 100.53
070315 0634.8641,S 10641.2196,E 2 10 236.7,M, 32.50 100.53
070316 0634.8639,S 10641.2197,E 2 10 236.6,M, 32.50 100.20
070317 0634.8640,S 10641.2199,E 2 10 236.4,M, 32.50 100.31
070318 0634.8640,S 10641.2198,E 2 10 236.5,M, 32.50 100.53
070319 0634.8641,S 10641.2197,E 2 10 236.6,M, 32.50 100.42
070320 0634.8641,S 10641.2196,E 2 10 236.6,M, 32.50 100.42
070321 0634.8642,S 10641.2195,E 2 10 236.6,M, 32.50 100.20
070322 0634.8642,S 10641.2195,E 2 10 236.6,M, 32.50 100.09
070323 0634.8642,S 10641.2196,E 2 10 236.5,M, 32.50 100.31
070324 0634.8641,S 10641.2196,E 2 10 236.6,M, 32.50 100.42
070325 0634.8641,S 10641.2195,E 2 10 236.6,M, 32.50 100.42
070326 0634.8641,S 10641.2195,E 2 10 236.5,M, 33.00 100.42
070327 0634.8640,S 10641.2194,E 2 10 236.2,M, 33.00 100.42
070328 0634.8640,S 10641.2193,E 2 10 236.4,M, 33.00 100.31
070329 0634.8640,S 10641.2193,E 2 10 236.4,M, 33.00 100.31
070330 0634.8640,S 10641.2193,E 2 10 236.4,M, 33.00 100.42
Connected 0:02:18 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Gambar 3.21. Hasil porting GPS Garmin 18LVC, sensor DS18B20 dan MPX4115A ke H8 untuk format GPGGA

BAB 4

ANALISIS HASIL PENELITIAN

4.1 Hasil Perbandingan Pengukuran Temperatur DS18B20 dan *Thermal Chamber* Vötsch VC³ 4018

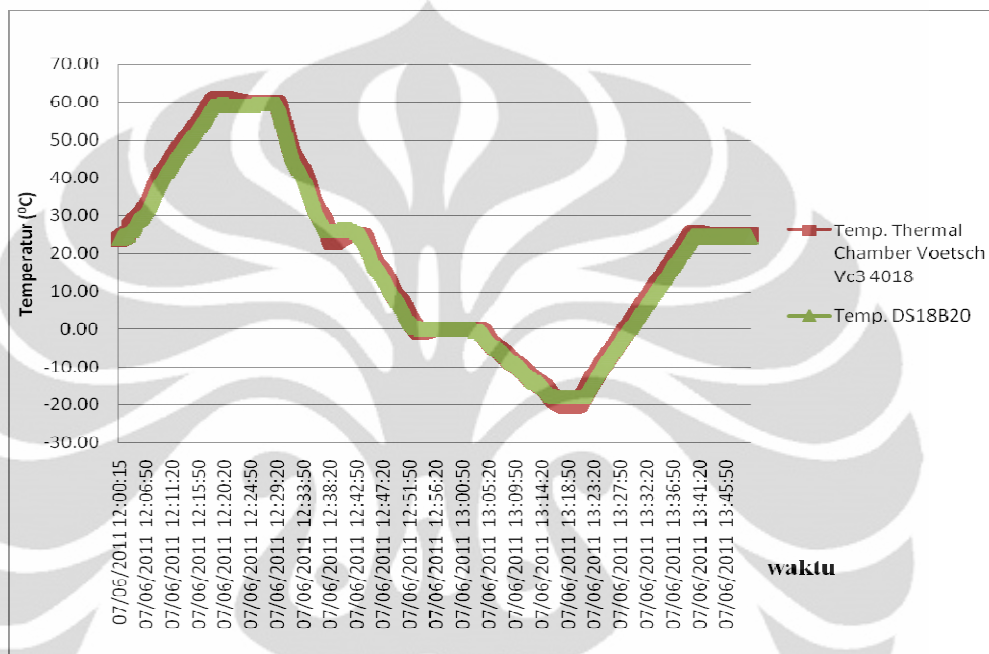
Proses kalibrasi DS18B20 dilakukan dengan membandingkan hasil temperatur yang terbaca oleh sensor dengan temperatur target (*set point*) pada *thermal chamber*. *Thermal chamber* yang digunakan sebagai pembanding adalah *thermal chamber* Vötsch VC³ 4018 yang memiliki rentang temperatur -40°C sampai +140°C. *Thermal chamber* ini terdiri atas 3 rak, sehingga memudahkan untuk peletakan sensor DS18B20 (Gambar 4.1).



Gambar 4.1. *Thermal chamber* Vötsch VC³ 4018

Selama proses kalibrasi berlangsung, setiap 6 menit sekali, temperatur target diubah per satu derajat celcius. Sebelum proses kalibrasi dilakukan, sensor DS18B20 terlebih dahulu diposisikan pada rak tengah dari *thermal chamber* Vötsch VC³ 4018. Setelah itu, temperatur target awal diatur pada 25°C. Kemudian temperatur dinaikkan hingga mencapai 60°C. Setelah mencapai 60°C, temperatur diturunkan hingga -20°C. Proses kalibrasi diakhiri dengan menaikkan temperatur dari -20°C hingga kembali ke temperatur awal, yakni pada saat 25°C.

Berikut ini adalah perbandingan hasil temperatur yang terbaca oleh sensor DS18B20 dengan temperatur target yang diatur pada *Thermal Chamber* Vötsch VC³ 4018.



Gambar 4.2. Hasil perbandingan pengukuran temperatur dari DS18B20 dan *thermal chamber* Vötsch VC³ 4018

Dari grafik diketahui, bahwa terdapat perbedaan temperatur yang terbaca oleh DS18B20 dengan hasil temperatur target dari *thermal chamber* Vötsch VC³ 4018. Beda temperatur terukur ini dapat dilihat pada Tabel 4.1.

Tabel 4.1. Beda temperatur terukur DS18B20 dengan *Thermal Chamber* Vötsch VC³ 4018 tiap 6 menit

No	Temperatur rata-rata per 6 menit (°C)
1	-0.4
2	-0.2
3	-0.5
4	-1.3
5	-0.4
6	-0.2
7	1.5

8	-0.1
9	0.1
10	0.1
11	0.0
12	0.2
13	0.8
14	0.9
15	0.1
16	-0.2
17	-0.5
18	-0.5
19	-0.5

Akurasi untuk sensor DS18B20 adalah $\pm 0.5^{\circ}\text{C}$ pada rentang -10°C hingga $+85^{\circ}\text{C}$. Namun, dari Gambar 4.2 dan Tabel 4.1, diketahui bahwa terdapat beda temperatur yang melebihi $\pm 0.5^{\circ}\text{C}$. Anomali ini pertama kali terjadi pada temperatur 60°C dengan beda temperatur mencapai -1.3°C . Kedua, pada penurunan temperatur mulai dari 28°C hingga 25°C dengan beda temperatur mencapai 1.5°C . Ketiga terjadi pada kenaikan temperatur mulai dari 18°C hingga 25°C .

Besarnya beda temperatur yang melebihi akurasi sensor DS18B20 untuk rentang -10°C hingga $+60^{\circ}\text{C}$, dapat dikarenakan *error* yang terjadi pada proses komunikasi *1-wire*. Pembacaan temperatur dan konversinya dilakukan berturut-turut berdasarkan protokol BEh dan 44h. Terdapat kemungkinan, *convert T master* menganggap proses pengkonversian telah selesai sebelum waktunya, sehingga DS18B20 mengirimkan bit 1 ke mikrokontroler H8/3069F. Sedangkan ketika terjadi penurunan temperatur hingga -20°C , beda temperatur mencapai 0.8°C hingga 0.9°C . Hal ini dikarenakan, akurasi $\pm 0.5^{\circ}\text{C}$ hanya untuk rentang temperatur -10°C hingga $+85^{\circ}\text{C}$.

4.2 Analisis Hasil Akuisisi Data

Akuisisi data untuk temperatur dan tekanan udara dibagi menjadi 2 metode. Pengambilan data pertama dilakukan dengan memindahkan alat dengan variasi ketinggian. Sedangkan untuk pengambilan data kedua dilakukan dengan meletakkan alat pada ketinggian yang sama dan data diakuisi selama 7 hari berturut-turut.

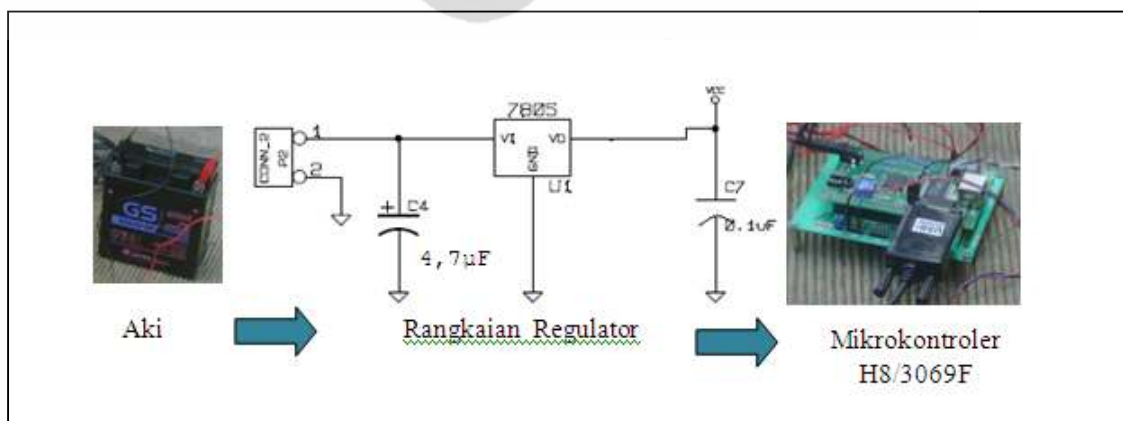
4.2.1 Analisis Hasil Akuisisi Data I

Akuisisi Data I dilakukan dengan memindahkan perangkat keras sistem akuisi data temperatur dan tekanan udara dengan perbedaan ketinggian. Oleh karena itu, sistem perangkat keras ini diletakkan di mobil. Semua sensor dan GPS diletakkan di atas mobil, sedangkan catu daya berupa aki dan mikrokontroler diletakkan di dalam mobil. Selanjutnya, seluruh akuisisi data ditampilkan di PC.



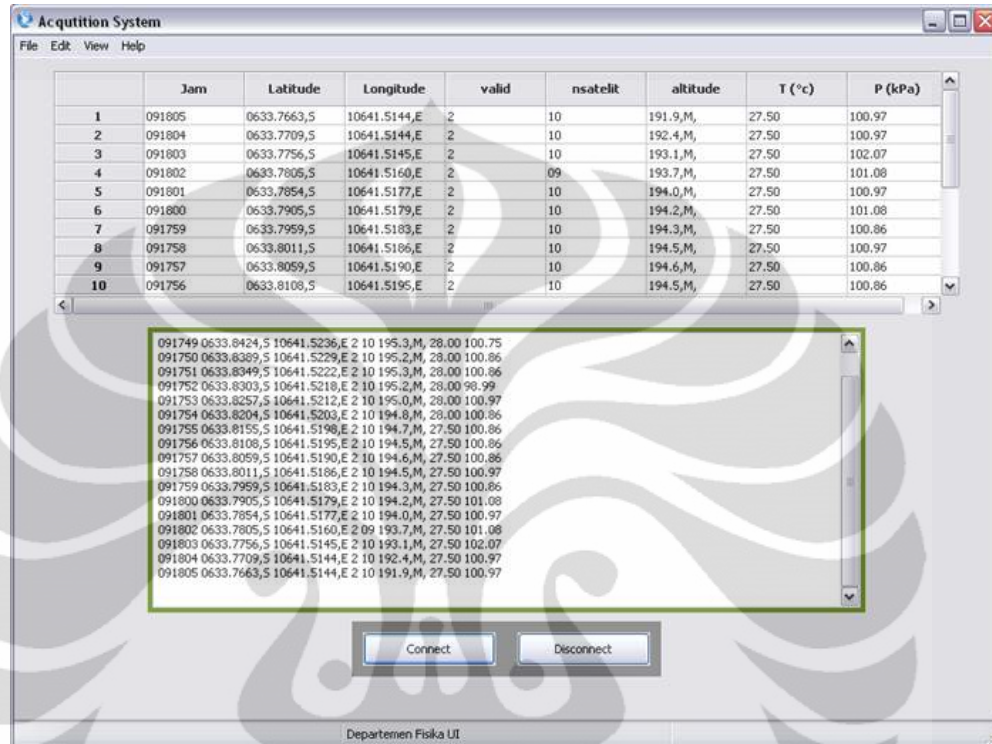
Gambar 4.3 Peletakan perangkat keras selama akuisisi (a) sensor-sensor dan GPS di atas mobil
(b) aki dan mikrokontroler H8/3069F di dalam mobil

Selama pengambilan data dilakukan, catu daya berasal dari aki 5Ah/12V. Karena mikrokontroler hanya membutuhkan tegangan input +5 V, maka diperlukan rangkaian regulator. IC yang digunakan pada rangkaian ini adalah LM7805 dengan tegangan output +5 V. Pada IC diberikan kapasitor *bypass* 4.7 μF pada pin 1 untuk mencegah osilasi dan untuk memperbaiki tanggapan transien diberi kapasitor *bypass* 0.1 μF pada pin 3 (Gambar 4.4).



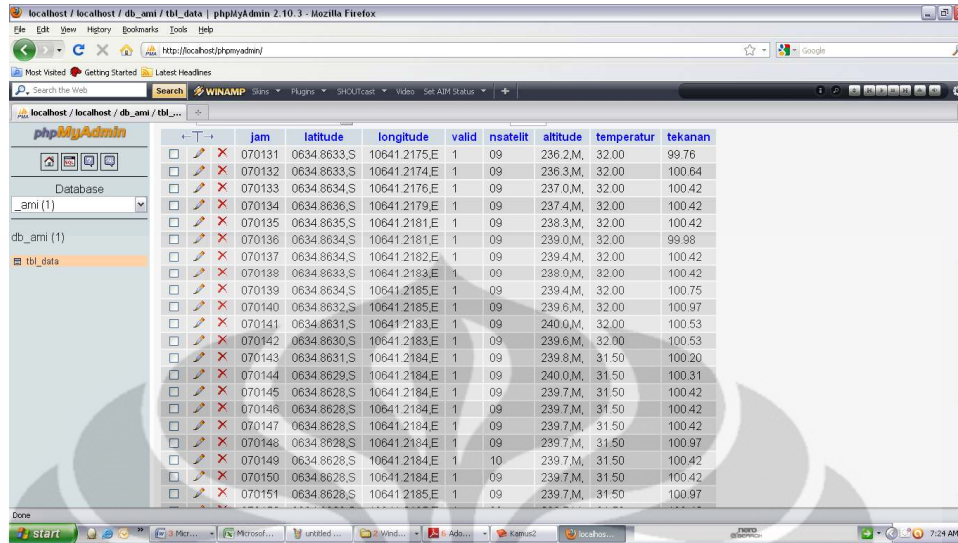
Gambar 4.4. Interkoneksi catu daya sistem perangkat keras

Akuisisi data I dilakukan pada tanggal 22 Mei 2011. Berikut ini adalah tampilan GUI dengan menggunakan bahasa pemrograman Python.



Gambar 4.5. GUI (*Graphical User Interface*) untuk sistem akuisisi berbasis Python

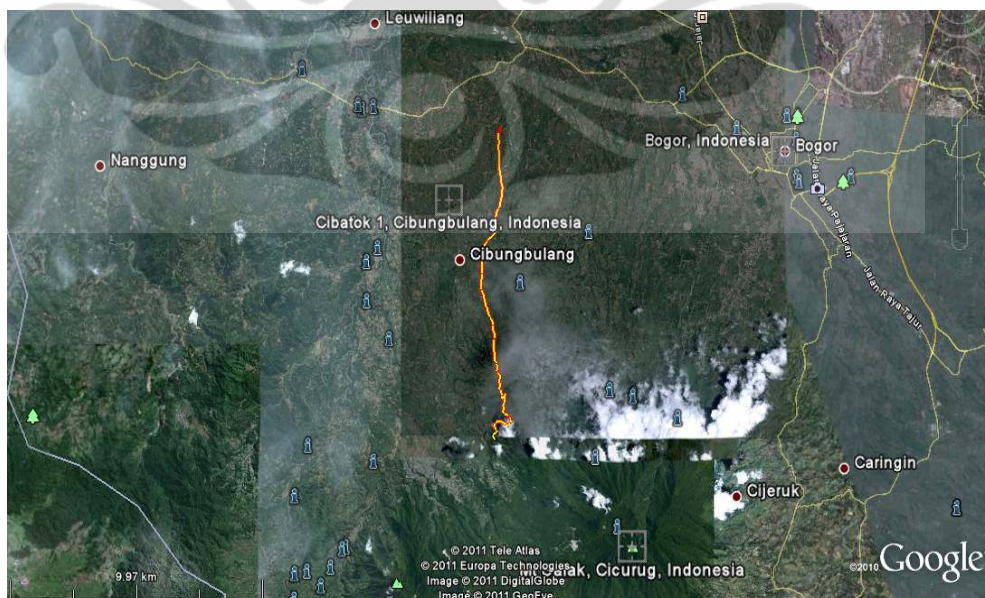
Hasil akuisisi data berturut-turut menampilkan berupa data waktu (UTC), posisi lintang dan bujur, validasi data GPS, jumlah satelit yang digunakan, ketinggian (m), temperatur (°C) dan tekanan udara (kPa). Sistem validasi dan informasi jumlah satelit ini bertujuan untuk memeriksa kebenaran data GPS yang memberikan informasi berupa waktu, koordinat, dan ketinggian. Apabila data validasi menunjukkan angka nol, maka data akuisisi akan diabaikan. Kemudian seluruh data selama akuisisi berlangsung disimpan dalam *data base* berbasis MySQL.



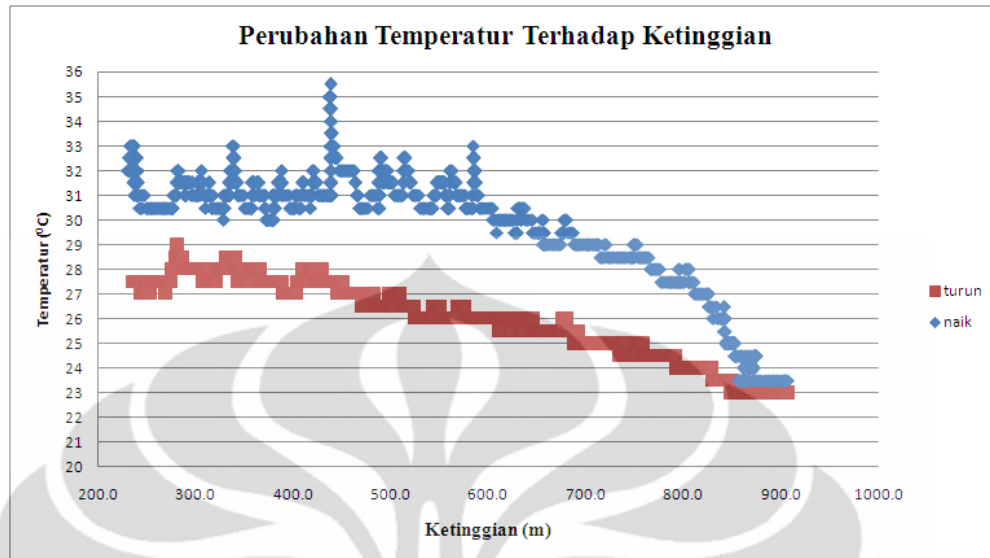
	jam	latitude	longitude	valid	nsatelit	altitude	temperatur	tekanan
<input type="checkbox"/>	070131	0634.8633,S	10641.2175,E	1	09	236,2,M	32,00	99,76
<input type="checkbox"/>	070132	0634.8633,S	10641.2174,E	1	09	236,3,M	32,00	100,64
<input type="checkbox"/>	070133	0634.8634,S	10641.2176,E	1	09	237,0,M	32,00	100,42
<input type="checkbox"/>	070134	0634.8636,S	10641.2179,E	1	09	237,4,M	32,00	100,42
<input type="checkbox"/>	070135	0634.8635,S	10641.2181,E	1	09	238,3,M	32,00	100,42
<input type="checkbox"/>	070136	0634.8634,S	10641.2181,E	1	09	239,0,M	32,00	99,98
<input type="checkbox"/>	070137	0634.8634,S	10641.2182,E	1	09	239,4,M	32,00	100,42
<input type="checkbox"/>	070138	0634.8633,S	10641.2183,E	1	00	238,0,M	32,00	100,42
<input type="checkbox"/>	070139	0634.8634,S	10641.2185,E	1	09	239,4,M	32,00	100,75
<input type="checkbox"/>	070140	0634.8632,S	10641.2185,E	1	09	239,6,M	32,00	100,97
<input type="checkbox"/>	070141	0634.8631,S	10641.2183,E	1	09	240,0,M	32,00	100,53
<input type="checkbox"/>	070142	0634.8630,S	10641.2183,E	1	09	239,6,M	32,00	100,53
<input type="checkbox"/>	070143	0634.8631,S	10641.2184,E	1	09	239,8,M	31,50	100,20
<input type="checkbox"/>	070144	0634.8629,S	10641.2184,E	1	09	240,0,M	31,50	100,31
<input type="checkbox"/>	070145	0634.8628,S	10641.2184,E	1	09	239,7,M	31,50	100,42
<input type="checkbox"/>	070146	0634.8628,S	10641.2184,E	1	09	239,7,M	31,50	100,42
<input type="checkbox"/>	070147	0634.8628,S	10641.2184,E	1	09	239,7,M	31,50	100,42
<input type="checkbox"/>	070148	0634.8628,S	10641.2184,E	1	09	239,7,M	31,50	100,97
<input type="checkbox"/>	070149	0634.8629,S	10641.2184,E	1	10	239,7,M	31,50	100,42
<input type="checkbox"/>	070150	0634.8628,S	10641.2184,E	1	09	239,7,M	31,50	100,42
<input type="checkbox"/>	070151	0634.8628,S	10641.2185,E	1	09	239,7,M	31,50	100,97

Gambar 4.6. Tampilan data base berbasis MySQL

Setelah diketahui latitude dan longitude, maka diubahlah satuan *degree minutes seconds* menjadi *degree decimal*. Hal ini dilakukan untuk membuat file .kml untuk melihat jalur akuisisi data dengan menggunakan mobil. Selanjutnya file .kml ditampilkan dengan menggunakan *software Google Earth*. Garis merah menunjukkan pengambilan data dari daerah rendah ke tinggi, sedangkan garis kuning untuk akuisisi data dari daerah tinggi ke rendah.

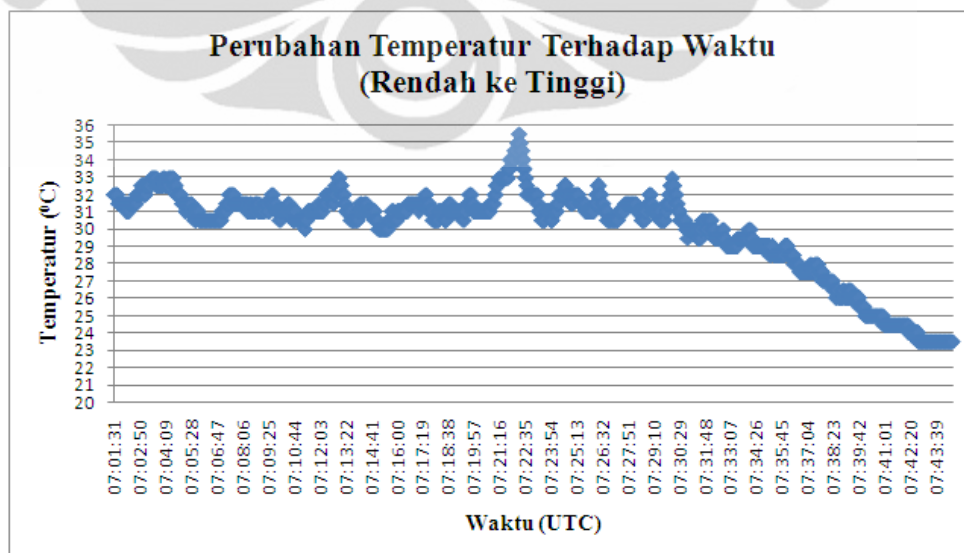


Gambar 4.7. Hasil pencitraan posisi selama akuisisi berlangsung

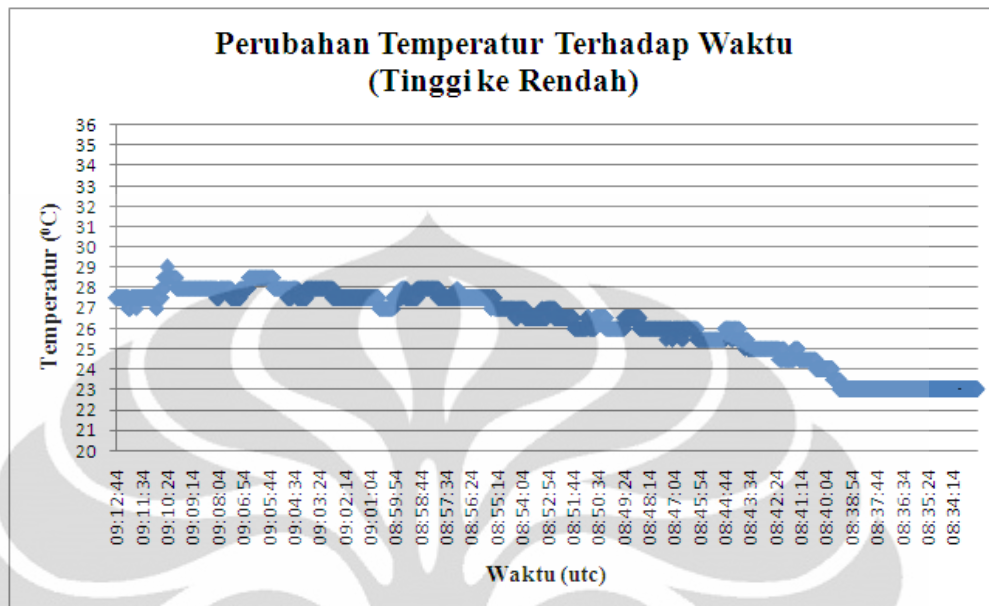


Gambar 4.8. Grafik perubahan temperatur terhadap ketinggian

Dari Gambar 4.8, titik berwarna biru menunjukkan pengambilan data dari daerah rendah ke tinggi. Ketinggian mula-mula untuk pengambilan data akuisisi adalah 236.2 m dan berakhir pada ketinggian 907.1 m. Temperatur awal yang terukur saat itu adalah 32°C, sedangkan temperatur pada ketinggian 907.1 m adalah 23.5°C. Pengambilan data dari daerah tinggi ke rendah ditunjukkan dengan titik berwarna merah. Ketinggian awal adalah 906.9 m dengan temperatur 23°C dan berakhir pada ketinggian 235.9 m dengan temperatur 27.5°C.



Gambar 4.9. Grafik perubahan temperatur terhadap waktu dari daerah rendah ke daerah tinggi



Gambar 4.10. Grafik perubahan temperatur terhadap waktu dari daerah tinggi ke daerah rendah

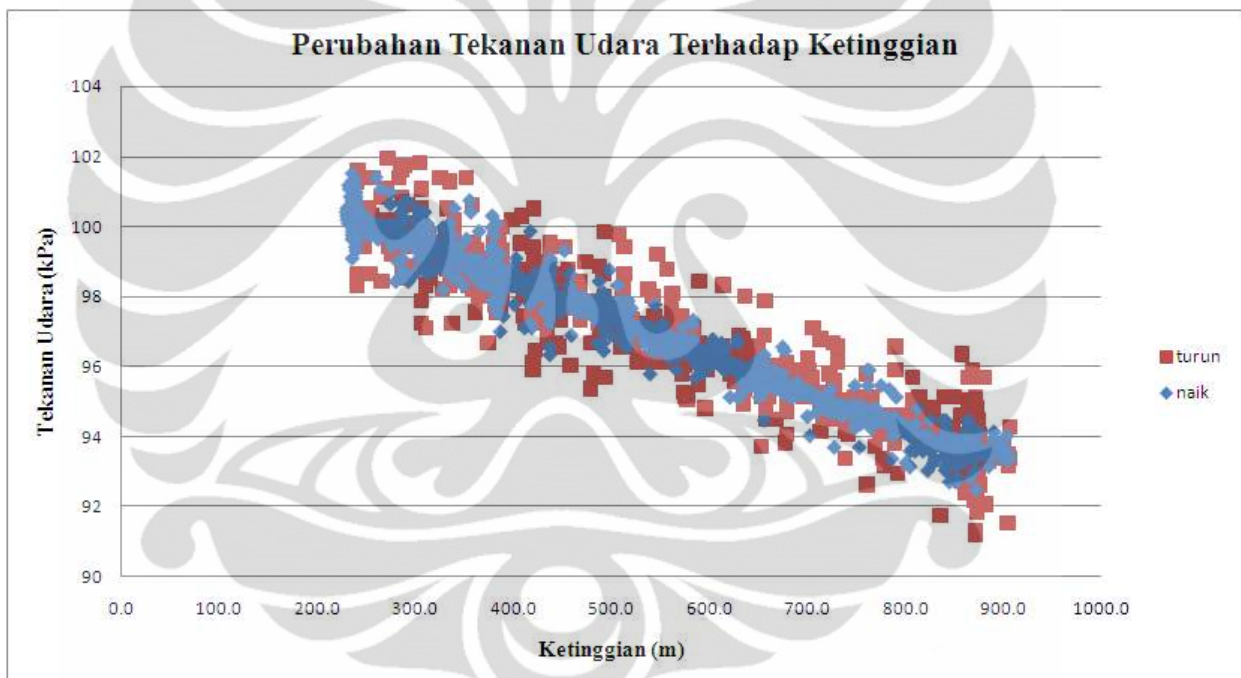
Berdasarkan Gambar 4.8, pada ketinggian yang sama namun dengan metode pengukuran yang berbeda, ternyata temperatur yang terukur bisa berbeda. Perbedaan temperatur terhadap waktu dapat dilihat pada Gambar 4.9 dan 4.10. Seperti pada ketinggian 900.7 m, diketahui temperatur saat 07:44:32 (UTC) adalah 23.5°C. Sedangkan pada pengambilan data dari daerah tinggi ke rendah, temperatur pada ketinggian 906.9 m adalah 23°C yang terukur pada 08:33:26 (UTC).

Sebaliknya ketika berada pada posisi daerah rendah (236.2 m), hasil akuisisi data temperatur dari rendah ke tinggi, temperatur awal adalah 32°C pada 07:01:31 (UTC). Sedangkan hasil akuisisi dari tinggi ke rendah, temperatur pada ketinggian paling rendah (235.9 m) adalah 27.5°C pada 09:12:44 (UTC). Adanya perbedaan temperatur pada ketinggian yang sama namun terukur pada waktu yang berbeda dapat disebabkan karena perbedaan lama penyinaran matahari dan banyak sedikitnya awan.

Hasil akuisisi data dari daerah rendah ke tinggi, pada waktu 07:01:31 (UTC) dengan ketinggian 236.2 m hingga 07:38:59 (UTC) dengan ketinggian 841.4 m, temperatur turun 5.5°C atau dengan gradien -0,009 °C/meter atau -0,002 °C/detik. Pada waktu 07:39:00 (UTC) dengan ketinggian 841.5 m hingga 07:44:32

(UTC) dengan ketinggian 907.1 m, temperatur turun 2.5°C atau dengan gradien $-0.04^{\circ}\text{C}/\text{meter}$ atau $-0.012^{\circ}\text{C}/\text{detik}$.

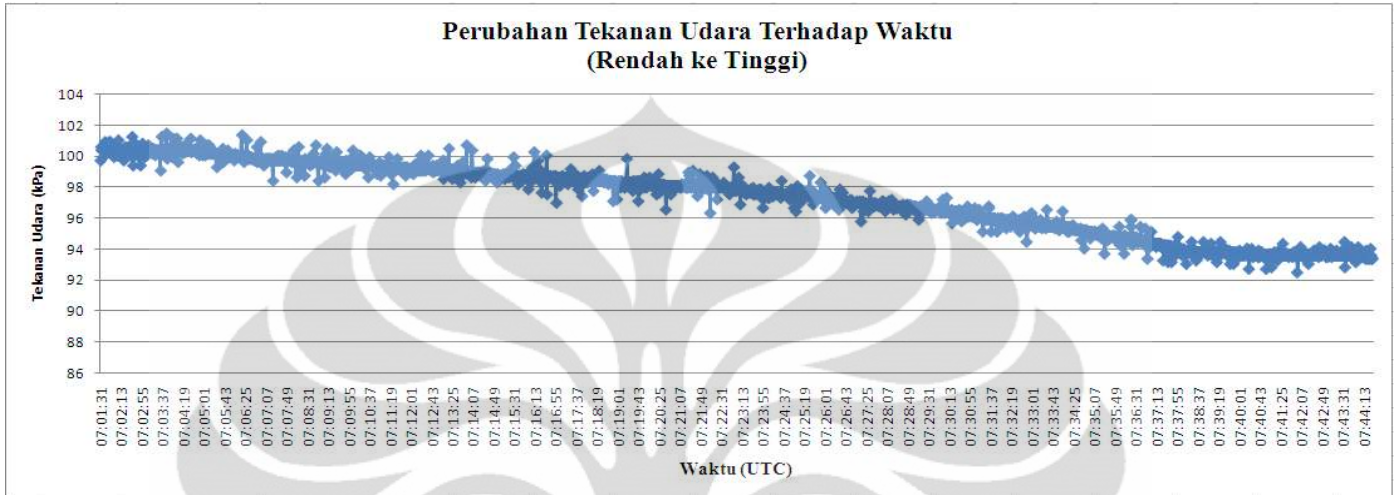
Sedangkan hasil akuisisi untuk daerah tinggi ke rendah, pada waktu 08:54:17 (UTC) dengan ketinggian 471.9 m dengan temperatur 26.5°C hingga 08:58:52 (UTC) dengan ketinggian 415.7 m dengan temperatur 28°C , temperatur turun 1.5°C atau dengan gradien $-0.03^{\circ}\text{C}/\text{meter}$ atau $-0.05^{\circ}\text{C}/\text{detik}$. Pada waktu 08:54:18 (UTC) dengan ketinggian 471.4 m dengan temperatur 26.5°C hingga 09:12:44 (UTC) dengan ketinggian 235.9 m dengan temperatur 27.5°C , temperatur turun 1°C atau dengan gradien $-0.004^{\circ}\text{C}/\text{meter}$ atau $-0.0009^{\circ}\text{C}/\text{detik}$.



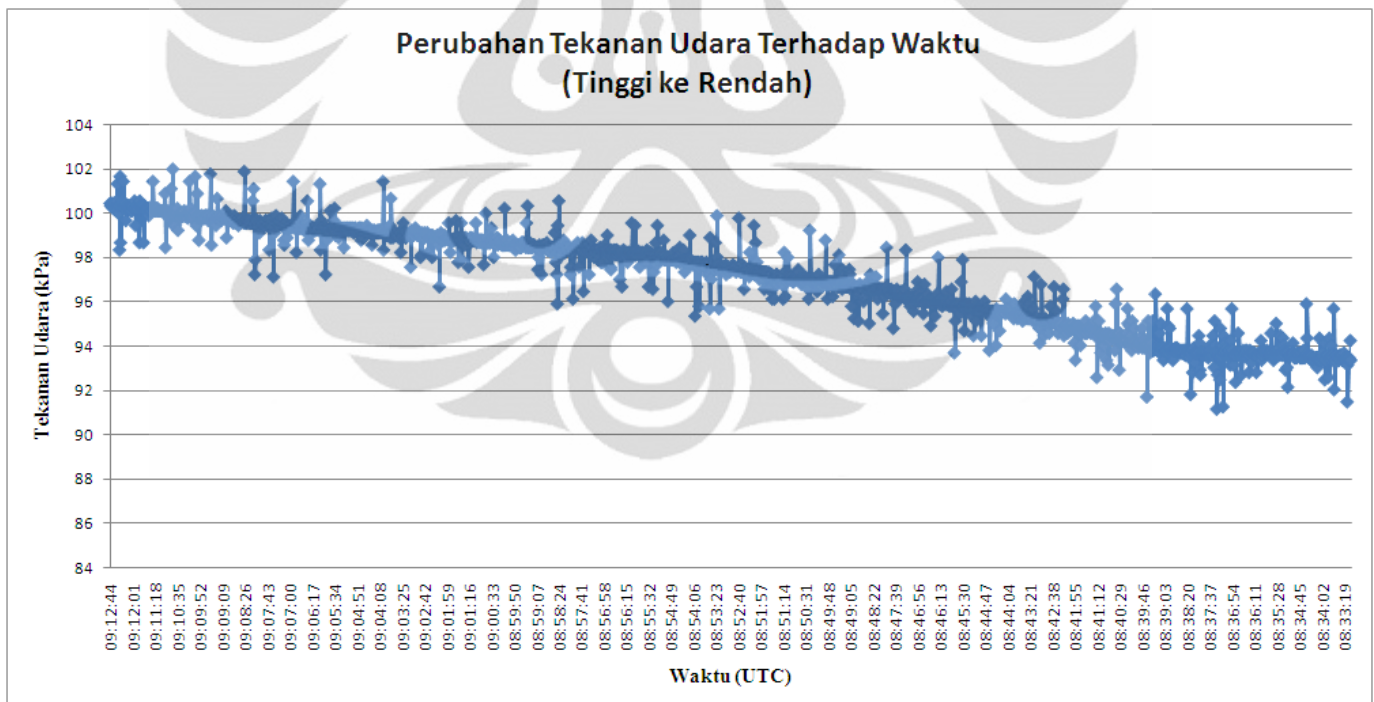
Gambar 4.11. Grafik perubahan tekanan udara terhadap ketinggian

Dari Gambar 4.11, titik berwarna biru menunjukkan pengambilan data dari daerah rendah ke tinggi. Ketinggian mula-mula untuk pengambilan data akuisisi adalah 236.2 m dan berakhir pada ketinggian 907.1 m. Tekanan udara awal yang terukur saat itu adalah 99.76 kPa, sedangkan temperatur pada ketinggian 907.1 m adalah 93.38 kPa. Pengambilan data dari daerah tinggi ke rendah ditunjukkan dengan titik berwarna merah. Ketinggian awal adalah 906,9 m

dengan tekanan udara 93.38 kPa dan berakhir pada ketinggian 235,9 m dengan tekanan udara 100.42 kPa.



Gambar 4.12. Grafik perubahan tekanan udara terhadap waktu dari daerah rendah ke daerah tinggi



Gambar 4.13. Grafik perubahan tekanan udara terhadap waktu dari daerah tinggi ke daerah rendah

Pada akuisisi data dari rendah ke tinggi, mulai dari 07:01:31 (UTC) pada ketinggian 236,2 m hingga 07:38:08 (UTC) pada ketinggian 819,9 m, tekanan

udara turun 5,72 kPa atau dengan gradien -0,003 kPa/detik atau -0,01 kPa/meter. Pada waktu 07:38:09 (UTC) pada ketinggian 820.6 m hingga 07:44:32 (UTC) pada ketinggian 907.1 m, tekanan udara turun 0.55 kPa atau dengan gradien -0.006 kPa/meter atau -0.001 kPa/detik.

Sedangkan akuisisi data dari daerah tinggi ke rendah, yaitu mulai dari 08:33:06 (UTC) dengan ketinggian 906.9 m hingga 08:40:06 (UTC) dengan ketinggian 814 m, tekanan udara turun 0.44 kPa atau dengan gradien -0.005 kPa/meter atau -0.001 kPa/detik. Pada waktu 08:40:07 (UTC) pada ketinggian 813,7 m hingga 09:12:44 (UTC) pada ketinggian 235,9 m, tekanan udara turun 5,94 kPa atau dengan gradien -0,003 kPa/detik atau -0,01 kPa/meter.

4.2.2 Analisis Hasil Akuisisi Data II

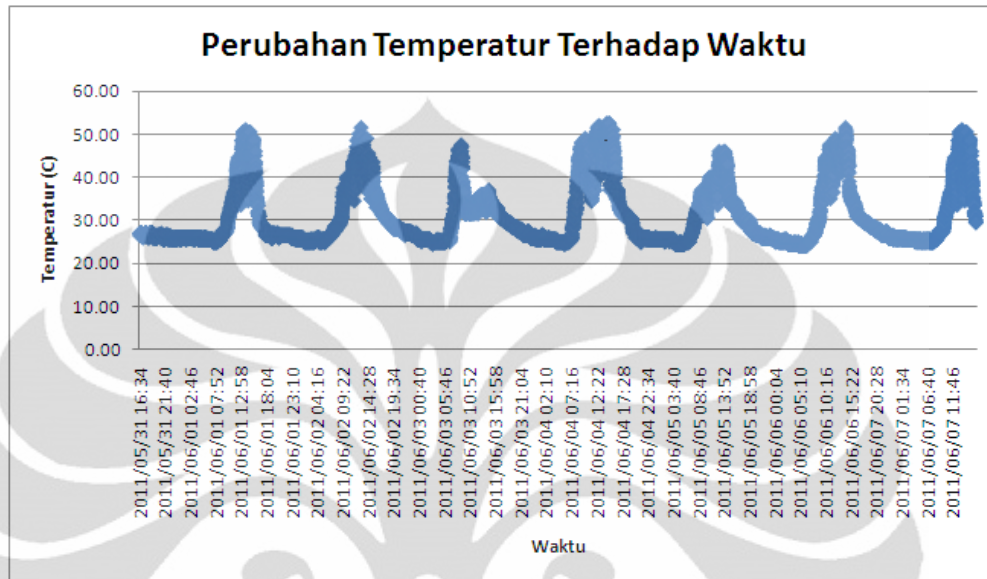
Hasil akuisisi data I memperlihatkan bahwa pengambilan data pada waktu yang berbeda namun dengan ketinggian yang sama, menunjukkan terdapat perubahan temperatur dan tekanan udara. Oleh karena itu, dilakukan pengambilan data temperatur dan tekanan udara selama 7 hari berturut-turut. Pada akuisisi data II, perangkat keras akuisisi data tidak *mobile* melainkan diletakkan pada ketinggian tertentu dari permukaan laut (191 m). Berikut ini adalah gambar untuk peletakan alat yang diam.



Gambar 4.14. Peletakan sensor temperatur dan tekanan udara di luar rumah

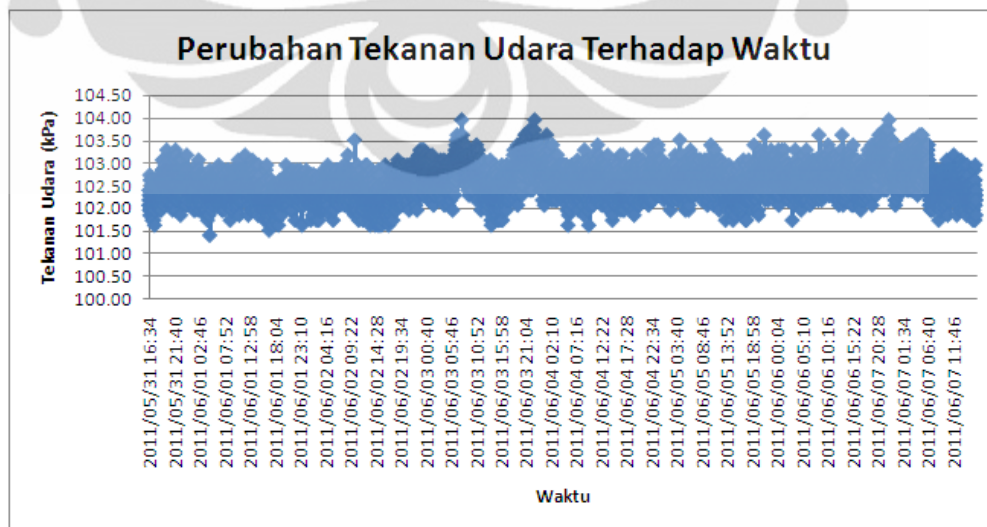
Hasil akuisisi data temperatur dan tekanan udara selama 7 hari menunjukkan bahwa, perubahan temperatur bersifat periodik dari hari ke hari. Temperatur paling rendah terjadi pada pukul 03:00 WIB hingga 05:00 WIB,

sedangkan temperatur paling tinggi terjadi pada pukul 12:00 WIB hingga 14:00 WIB (Gambar 4.15).



Gambar 4.15. Grafik perubahan temperatur terhadap waktu

Sedangkan perubahan tekanan udara selama 7 hari, tidak menunjukkan perubahan tekanan udara yang signifikan. Rata-rata tekanan udara yang terakuisisi adalah 102.58 kPa (Gambar 4.16).



Gambar 4.16. Grafik perubahan tekanan udara terhadap waktu

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Sensor DS18B20 bekerja dengan baik dengan resolusi 9 bit untuk ketinggian 235.9 m hingga 399.8 m.
2. Sensor MPX4115A bekerja dengan baik untuk ketinggian 235.9 m hingga 906.9 m.
3. Hasil akuisisi data temperatur dari daerah rendah ke tinggi, pada ketinggian 236.2 m hingga 841.4 m, temperatur turun 5.5°C atau dengan gradien $-0.009^{\circ}\text{C}/\text{meter}$ dan pada ketinggian 841.5 m hingga 907.1 m, temperatur turun 2.5°C atau dengan gradien $-0.04^{\circ}\text{C}/\text{meter}$.
4. Hasil akuisisi data temperatur untuk daerah tinggi ke rendah, pada ketinggian 471.9 m hingga 415.7 m, temperatur turun 1.5°C atau dengan gradien $-0.03^{\circ}\text{C}/\text{meter}$ dan pada ketinggian 471.4 m hingga 235.9 m, temperatur turun 1°C atau dengan gradien $-0.004^{\circ}\text{C}/\text{meter}$.
5. Hasil akuisisi data tekanan udara dari rendah ke tinggi, mulai dari ketinggian 236.2 m hingga 819.9 m, tekanan udara turun 5.72 kPa atau dengan gradien $-0,01\text{ kPa}/\text{meter}$ dan pada ketinggian 820.6 m hingga 907.1 m, tekanan udara turun 0.55 kPa atau dengan gradien $-0.006\text{ kPa}/\text{meter}$.
6. Hasil akuisisi data tekanan udara dari daerah tinggi ke rendah, yaitu mulai dari ketinggian 906.9 m hingga 814 m, tekanan udara turun 0.44 kPa atau dengan gradien $-0.005\text{ kPa}/\text{meter}$ dan pada ketinggian 813.7 m hingga 235.9 m, tekanan udara turun 5.94 kPa atau dengan gradien $-0,01\text{ kPa}/\text{meter}$.
7. Pada ketinggian yang tetap (191 m), perubahan temperatur terhadap waktu menunjukkan perubahan yang periodik dari hari ke hari. Sedangkan tekanan udara cenderung tetap dengan rata-rata 102.58 kPa.
8. Hubungan antara tekanan udara dengan ketinggian lebih rumit, karena kerapatan gas tidak konstan tetapi tergantung pada tekanan. Oleh

karena itu, pada penelitian ini tidak dicari hubungan antara temperatur dan tekanan udara.

9. Akuisisi pada sistem ini belum menerapkan telemetri.

5.2 SARAN

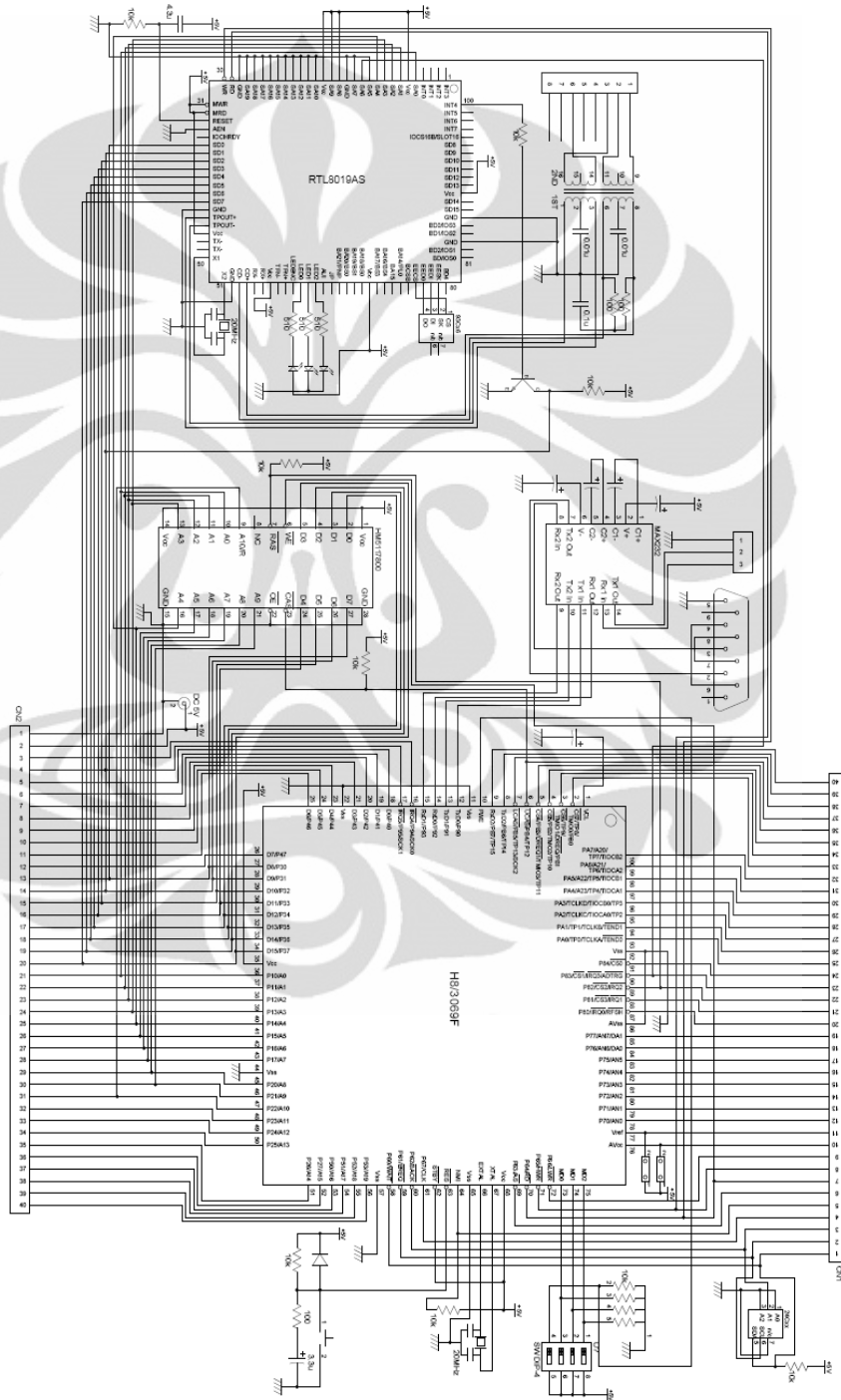
1. Resolusi DS18B20 untuk akuisisi data temperatur sebaiknya menggunakan resolusi 12 bit untuk mendapatkan data yang lebih teliti.
2. Penelitian lebih lanjut untuk pengembangan dan menambah kualitas penelitian ini, yaitu dengan menerapkan *software* pemantauan temperatur dan tekanan udara pada OS yang *open source*, seperti Linux.
3. Penerapan komunikasi nirkabel pada transmisi sinyal dari sensor/*receiver* ke *processor* (mikrokontroler) yang bermanfaat dalam akuisisi data jarak jauh.

DAFTAR ACUAN

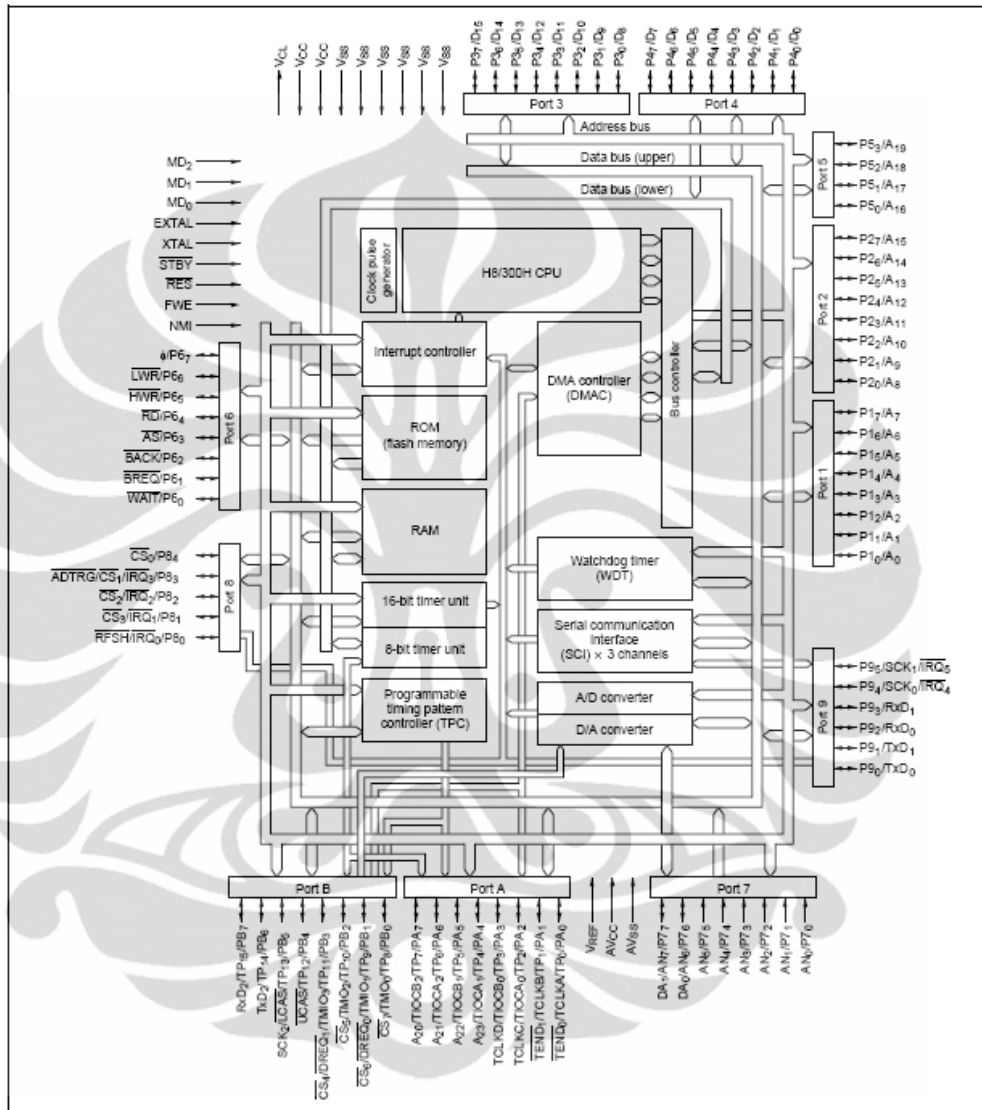
- [1] Kodoatie, Robert J. dan Roestam Syarief. 2010. *Tata Ruang Air*. C.V Andi Offset, Yogyakarta.
- [2] Fujisawa, Yukiho. 2003. *The Introduction to The H8 Microcontroller*. Ohmsha Ltd., Tokyo, Japan.
- [3] Berlin, Peter. 2004. *“The Geostationary Applications Satellite”*. Cambridge University Press, Inggris.
- [4] Tipler, Paul A. 1991. *“Physics for Scientist and Engineers, Third Edition”*. Worth Publisher, Inc.
- [5] Dalas semiconductor “*DS18B20 Programmable Resolution 1 wire Digital Thermometer*”, diakses dari <http://pdfserv.maxim-ic.com/en/ds/DS18B20.pdf> pada tanggal 28 Desember 2010 pukul 08.00 WIB
- [6] MPX4115A Datasheet.
- [7] N. Ingot Marito. 2008. *Sistem Navigasi Helikopter Berdasarkan Data Posisi Secara Telemetri*. Teknik Elektro, Universitas Indonesia, Depok.
- [8] Garmin. 2005. *GPS 18 Technical Specifications*. Garmin International Inc., USA.
- [9] Tardi, Muh. Ibnu Malik. 2009. *Aneka Proyek Mikrokontroler PIC16F84/A. Rakit Segera 13 Proyek Menarik dengan Mikrokontroler PIC16F84/A*. PT Elex Media Komputindo, Jakarta.
- [10] Renesas Solutions Corp. 2005. *H8/3068F-ZTAT™ Hardware Manual*. Renesas Technology Corp., Japan.
- [11] Purwadi, Hari SST, dkk. 2008. *Prototype Sistem Telemetri untuk Pengukuran Arus Beban 3 Phasa melalui Kanal Radio*. Politeknik Negeri Samarinda, Samarinda.
- [12] Santoso, Berkah I. 2010. *Bahasa Pemrograman Python di Platform GNU/Linux*.
- [13] Nurliana, Rina, dkk. Aplikasi Database dengan PHP dan MySQL.
- [14] Tim H8, 2007, *Modul Praktikum Renesas H8/3069F*, Departemen Fisika.

LAMPIRAN A

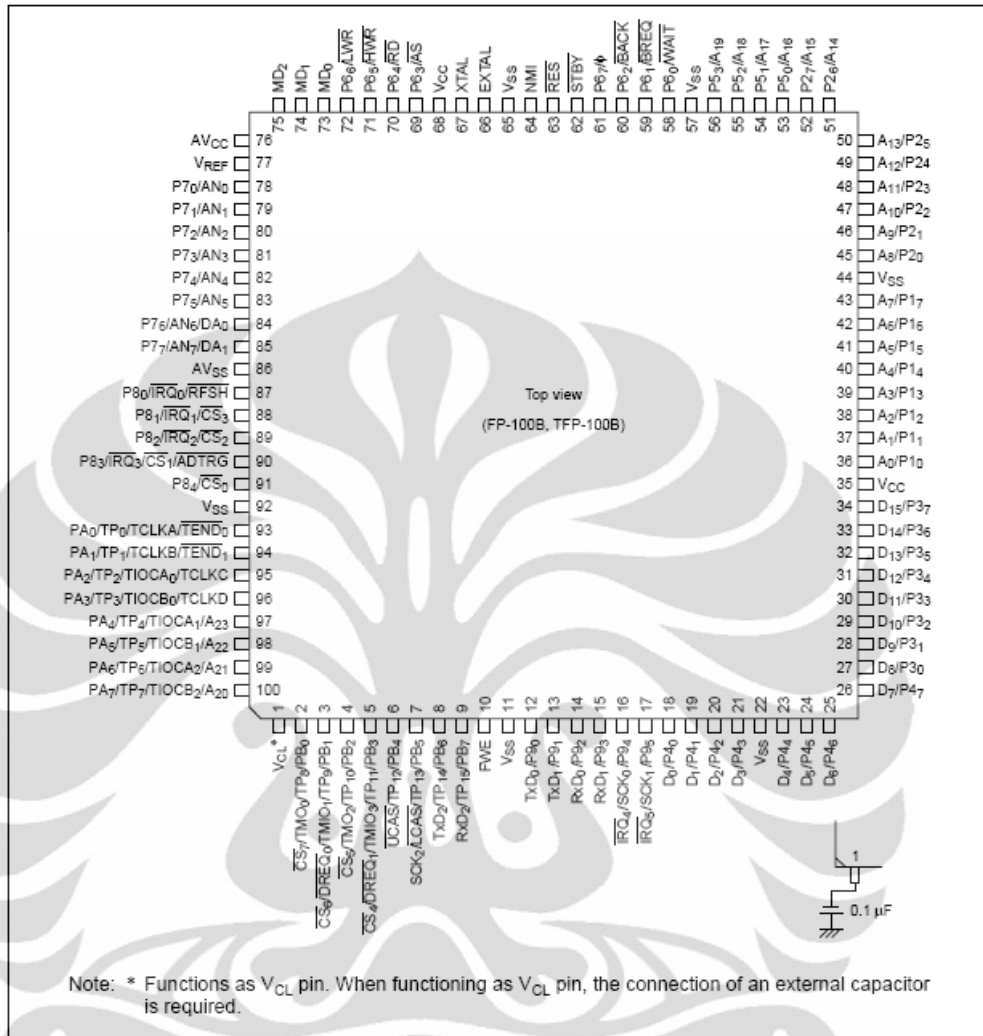
Mikrokontroler H8/3069 F



Gambar A.1 Minimum System H8/3069F



Gambar A.2 Blok diagram H8/3069F



Gambar A.3. Pin arrangement H8/3069

LAMPIRAN B

```
/*
main program untuk :
1. GPS garmin 18LVC
2. ds18b20
3. mPX4115a
*/

#include "ioh83069.h"
#include "uart.h"
#include "delay.h"

#include "ds1820.h"
#include "romcode.h"

#include "interrupt.h"
#include "adc.h"

#include <stdlib.h>
#include <stdio.h>

//untuk gps
char cmdbuf[512], data, waktu[8],latitude[14],longitude[14],
altitude[10], nsatelit[4], valid[4];
unsigned short i;
unsigned short cmdln;

unsigned char sp [SCP_LEN];
char buf [10];

unsigned short adcbuf[10];
char bufmpx[10];
unsigned short value;
int pressure;

void ISR_RXI0 (void)
{
    data = RDR0;
    SSR0 &= ~(1 << RDRF);

    if (cmdln == 0) {

        if (data == '$') {
            cmdbuf[cmdln] = data;
            cmdln++;
        } else {
            cmdln = 0;
        }

    } else {

        if (data != 13) {
```

```

        cmdbuf[cmdln] = data;
        cmdln++;

    } else {

        cmdbuf[cmdln] = '\0';

        if (cmdbuf[0] != '$') {

            cmdln = 0;

        } else {

            if ((cmdbuf[1] == 'G') && (cmdbuf[2] ==
                'P') && (cmdbuf[3] == 'G') && (cmdbuf[4] == 'G') && (cmdbuf[5] ==
                'A')) {

                //uart_str(&cmdbuf[0]);
                //Parsing waktu
                //waktu mulai pada posisi ke 7
                dengan panjang 6
                for (i = 0; i < 6; i++) {
                    waktu[i] = cmdbuf[7 + i];
                }
                i++;
                waktu[i] = '\0';

                //Parsing latitude
                //latitude mulai pada posisi ke 14
                dengan panjang 11
                for (i = 0; i < 11; i++) {
                    latitude[i] = cmdbuf[14 + i];
                }
                i++;
                latitude[i] = '\0';

                //Parsing longitude
                //longitude mulai pada posisi ke 26
                dengan panjang 12
                for (i = 0; i < 12; i++) {
                    longitude[i] = cmdbuf[26 + i];
                }
                i++;
                longitude[i] = '\0';

                //Parsing valid
                //longitude mulai pada posisi ke 39
                dengan panjang 1
                for (i = 0; i < 1; i++) {
                    valid[i] = cmdbuf[39 + i];
                }
                i++;
                valid[i] = '\0';
            }
        }
    }

```

```

//Parsing nsatelit
//longitude mulai pada posisi ke 41
dengan panjang 2
for (i = 0; i < 2; i++) {
    nsatelit[i] = cmdbuf[41 + i];
}
i++;
nsatelit[i] = '\0';

//Parsing altitude
//altitude mulai pada posisi ke 48
dengan panjang 8
for (i = 0; i < 8; i++) {
    altitude[i] = cmdbuf[48 + i];
}
i++;
altitude[i] = '\0';

uart_str(&waktu[0]);
uart_strP(" ");

uart_str(&latitude[0]);
uart_strP(" ");

uart_str(&longitude[0]);
uart_strP(" ");

uart_str(&valid[0]);
uart_strP(" ");

uart_str(&nsatelit[0]);
uart_strP(" ");

uart_str(&altitude[0]);
uart_strP(" ");

ds1820_convert_all();
ds1820_match_rom(&rom[0]);
ds1820_get_scratchpad(&sp[0]);
ds1820_print_suhu(&buf[0], &sp[0]);
uart_str(&buf[0]);
uart_strP(" ");

adc_getval_grp(&adcbuf[0], GRMPX);
//ambil data dari channel 4
adcbuf[0];
value = (adcbuf[1] << 8) |

pressure = (value*11) + 1055;
sprintf(&bufmpx[0], "%d.%02d",
(pressure/100), (pressure%100));
uart_str(&bufmpx[0]);
uart_strP("\r\n");
}

```



```
        cmdln = 0;
    }
}

int main (void)
{
    uart_init();
    uart_strP("Inisialisasi Uart\r\n");
    ds1820_init();
    uart_strP("Inisialisasi Ds18B20\r\n");
    adc_init();
    uart_strP("Inisialisasi Ds18B20\r\n");
    cmdln = 0;
    sei();
    while(1);
    return 0;
}
```



LAMPIRAN C

```
#MAIN PROGRAM untuk GUI

import wx, wx.grid
import os, time
import serial, threading

import db_tabel as tbl

#Import library untuk matplotlib dengan backend Wx
import matplotlib
matplotlib.use('WXAgg')
import numpy as np
from matplotlib.figure import Figure
from matplotlib.backends.backend_wxagg import FigureCanvasWxAgg as
Canvas
from matplotlib.backends.backend_wxagg import
NavigationToolbar2WxAgg as Toolbar
import pylab

#Definisi konstanta
MYCOM = 'com3' # definisi port serial
RETURN = chr(13)
CARRIAGE = chr(10)

#Definisi table
label_tab = ['Jam', 'Latitude',
'Longitude', 'valid', 'nsatelit', 'altitude', 'T ('+chr(176)+'c)', 'P
(kPa)']
num_rows = 18
num_cols = 8

SERIALRX = wx.NewEventType()
# bind to serial data receive events
EVT_SERIALRX = wx.PyEventBinder(SERIALRX, 0)

class SerialRxEvent(wx.PyCommandEvent):
    eventType = SERIALRX

    def __init__(self, windowID, data):
        wx.PyCommandEvent.__init__(self, windowID, self.eventType,
data)
        self.data = data

    def Clone(self):
        self.__class__(self.GetId(), self.data)

class Telecommand(wx.Frame):
    def __init__(self, parent, id):
        wx.Frame.__init__(self, parent, id, title='Acquisition
System', size=(900,676))
        color = wx.SystemSettings.GetColour(wx.SYS_COLOUR_BTNFACE)
```

```

        self.SetBackgroundColour(color)
        self.SetIcon(wx.Icon('network.ico',
wx.BITMAP_TYPE_ICO))

        #initial value
        self.ser = serial.Serial()
        self.ser.timeout = 0.5
        self.thread = None
        self.alive = threading.Event()
        self.data = []
        self.tbl_jam = []
        self.tbl_latitude = []
        self.tbl_longitude = []
        self.tbl_valid = []
        self.tbl_nsatelit = []
        self.tbl_altitude = []
        self.tbl_temperatur = []
        self.tbl_tekanan = []

        h1 = wx.Font(12, wx.SWISS, wx.NORMAL, wx.BOLD)

#===== panel-panel utama =====
        pnl1 = wx.Panel(self, -1)
        pnl2 = wx.Panel(self, -1)
        pnl2.SetBackgroundColour('medium forest green')
        pnl3 = wx.Panel(self, -1)
        pnl3.SetBackgroundColour('grey')

#===== widget panel 1=====
        self.grid = wx.grid.Grid(pnl1, -1, size=(820,220))
        self.grid.CreateGrid(num_rows, num_cols)

        for i in range(num_cols):
            self.grid.SetColSize(i, 91)

        for row in range(num_rows):
            for col in range(num_cols):
                self.grid.SetColLabelValue(col,
label_tab[col])

        border1 = wx.FlexGridSizer()
        border1.Add(self.grid, 1, wx.EXPAND | wx.ALL, 4)
        pnl1.SetSizer(border1)

#===== widget panel 2=====
        self.log = wx.TextCtrl(pnl2, -1, '', size=(642, 250),
style = wx.TE_MULTILINE | wx.TE_PROCESS_ENTER)
        sizer2 = wx.BoxSizer(wx.VERTICAL)
        sizer2.Add(self.log, 1, wx.EXPAND | wx.ALL, 4)
        pnl2.SetSizer(sizer2)

#=====widget panel 3 =====

        btn1 = wx.Button(pnl3, -1, 'Connect', size = (120,
30))

```

```

30))

btn2 = wx.Button(pnl3, -1, 'Disconnect', size = (120,

self.Bind(wx.EVT_BUTTON, self.OnBtn1, btn1)
self.Bind(wx.EVT_BUTTON, self.OnBtn2, btn2)

bbtn = wx.FlexGridSizer(0, 4, 0, 0)
bbtn.Add(btn1, 2, wx.EXPAND | wx.ALL, 10)
bbtn.Add(btn2, 2, wx.EXPAND | wx.ALL, 10)

sizer4 = wx.BoxSizer(wx.VERTICAL)
sizer4.Add(bbtn, 0, wx.EXPAND | wx.ALL, 0)

sizer5 = wx.BoxSizer()
sizer5.Add(sizer4, 1, wx.EXPAND | wx.ALL, 0)
pnl3.SetSizer(sizer5)

#=====penyatuan panel=====

box = wx.BoxSizer(wx.VERTICAL)
box.Add(pnl1, 0, wx.ALIGN_CENTER_HORIZONTAL | wx.ALL,
4)
box.Add(pnl2, 0, wx.ALIGN_CENTER_HORIZONTAL | wx.ALL,
4)
box.Add(pnl3, 0, wx.ALIGN_CENTER_HORIZONTAL | wx.ALL,
4)

box2 = wx.BoxSizer()
box2.Add(box, 1, wx.EXPAND | wx.ALL, 4)

self.SetSizer(box2)
self.Layout()

#=====menu bar=====

menuBar = wx.MenuBar()

mfile = wx.Menu() #Isi dari mfile
self.saveMenuItem = mfile.Append(-1, '&Save', 'Save')
self.saveMenuItem = mfile.Append(-1, '&Print',
'Print')
mfile.AppendSeparator()
self.exitMenuItem = mfile.Append(-
1,"E&xit\tAlt+X","Exit From software")
medit = wx.Menu() #Isi dari medit
self.connectMenuItem = medit.Append(-1, '&Copy',
'Copy')

mview = wx.Menu() #Isi dari mview
self.saveMenuItem = mview.Append(-1, '&Zoom In', 'Zoom
In')
self.saveMenuItem = mview.Append(-1, '&Zoom Out',
'Zoom Out')

mhhelp = wx.Menu() #Isi dari mhhelp

```

```

        self.helpMenuItem = mhelp.Append(-1, "&Help\tF1", "Help")
        mhelp.AppendSeparator()
        self.aboutMenuItem = mhelp.Append(-1, "&About", "About")

        #Definisi dari Menubar
        menuBar.Append(mfile, '&File')
        menuBar.Append(medit, '&Edit')
        menuBar.Append(mview, '&View')
        menuBar.Append(mhelp, '&Help')
        self.SetMenuBar(menuBar)

#=====status bar=====
self.statusbar = self.CreateStatusBar(3,
wx.ST_SIZEGRIP)
self.statusbar.SetStatusWidths([-1, -1, -1])

        statusbar_fields = [("", "Departemen Fisika UI"),
        ("")]

        for i in range(len(statusbar_fields)):
            self.statusbar.SetStatusText(statusbar_fields[i], i)

#=====Event Serial=====
        self.Bind(EVT_SERIALRX, self.OnSerialRead)

        def StartThread(self):
            """Start the receiver thread"""
            self.thread =
            threading.Thread(target=self.ComPortThread)
            self.thread.setDaemon(1)
            self.alive.set()
            self.thread.start()

        def StopThread(self):
            """Stop the receiver thread, wait until it's
            finished."""
            if self.thread is not None:
                self.alive.clear() #clear alive event
            for thread
            self.thread.join() #wait until thread
            has finished
            self.thread = None

        def ComPortThread(self):
            """Thread that handles the incoming traffic. Does the
            basic input
            transformation (newlines) and generates an
            SerialRxEvent"""
            while self.alive.isSet(): #loop while
            alive event is true
                text = self.ser.read(1) #read one,
            with timeout
                if text: #check
            if not timeout
                n = self.ser.inWaiting() #look if
            there is more to read

```

```

        if n:
            text = text + self.ser.read(n)
#get it
            text = text.replace('\r', '')
            #text = text.replace('\r', '\n')

            event = SerialRxEvent(self.GetId(), text)

self.GetEventHandler().AddPendingEvent(event)

def OnSerialRead(self, event):
    """Handle input from the serial port."""
    text = event.data

    data_mentah = []
    data_jam = ''
    data_latitude = ''
    data_longitude = ''
    data_valid = ''
    data_nsatelit = ''
    data_altitude = ''
    data_temperatur = ''
    data_tekanan = ''

    self.data.append(text)

    if self.data[-1] == '\n' and len(self.data) > 7:
        temp = ''.join(self.data)
        data_mentah = temp.split(' ')
        self.data = []

        if data_mentah > 8:
            data_jam = data_mentah[0]
            data_latitude = data_mentah[1]
            data_longitude = data_mentah[2]
            data_valid = data_mentah[3]
            data_nsatelit = data_mentah[4]
            data_altitude = data_mentah[5]
            data_temperatur = data_mentah[6]
            data_tekanan = data_mentah[7]

            self.tbl_jam.append(data_mentah[0])
            self.tbl_latitude.append(data_mentah[1])
            self.tbl_longitude.append(data_mentah[2])
            self.tbl_valid.append(data_mentah[3])
            self.tbl_nsatelit.append(data_mentah[4])
            self.tbl_altitude.append(data_mentah[5])
            self.tbl_temperatur.append(data_mentah[6])
            self.tbl_tekanan.append(data_mentah[7])

        for i in range(num_rows):
            try:
                self.grid.SetCellValue(i, 0,
                    '%s' % self.tbl_jam[-(i+1)])
                self.grid.SetCellValue(i, 1,
                    '%s' % self.tbl_latitude[-(i+1)])

```

```

        self.grid.SetCellValue(i, 2,
'%s' % self.tbl_longitude[-(i+1)])
        self.grid.SetCellValue(i, 3,
'%s' % self.tbl_valid[-(i+1)])
        self.grid.SetCellValue(i, 4,
'%s' % self.tbl_nsatelit[-(i+1)])
        self.grid.SetCellValue(i, 5,
'%s' % self.tbl_altitude[-(i+1)])
        self.grid.SetCellValue(i, 6,
'%s' % self.tbl_temperatur[-(i+1)])
        self.grid.SetCellValue(i, 7,
'%s' % self.tbl_tekanan[-(i+1)])
    except IndexError:
        pass

def OnBtn1(self, event):
    self.log.WriteText('Connect to serial '+MYCOM+'...\n')

    try:
        self.ser = serial.Serial(MYCOM)
        self.StartThread()

    except serial.SerialException:
        warn = wx.MessageDialog(self, 'Tidak dapat
membuka port '+MYCOM+'...', 'ERROR !', wx.OK | wx.ICON_STOP)
        warn.ShowModal()
        warn.Destroy()

def OnBtn2(self, event):
    self.log.WriteText('Disconnect serial...\n')
    self.StopThread()
    self.ser.close()

```