



**UNIVERSITAS INDONESIA**

**SEISMIC DATA LOGGER TERSINKRONISASI WAKTU GPS  
BERBASIS MIKROKONTROLLER H8/3069 F**

**SKRIPSI**

**HARRIS SETYO FERNANTO**

**0606068240**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM 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 : HARRIS SETYO FERNANTO**

**NPM : 0606068240**

**Tanda Tangan :**

**Tanggal : 27 JUNI 2011**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : HARRIS SETYO FERNANTO  
NPM : 0606068240  
Program Studi : INSTRUMENTASI ELEKTRONIKA  
Judul Skripsi : SEISMİK DATA LOGGER TERSINKRONISASI  
WAKTU GPS BERBASIS MIKROKONTROLLER  
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 Pengetahuan Alam, Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : Dr. Eng. Supriyanto, M.Sc

Penguji I : Dr. B.E.F Da Silva, M.Sc

Penguji II : Dr. Tony Mulia

Ditetapkan di : Depok

Tanggal : 27 Juni 2011

## KATA PENGANTAR

Puji syukur hanya kepada Allah SWT yang selalu memberikan anugerah terindah-Nya kepada penulis dan telah menuntun penulis menuangkan tiap untaian kata sehingga skripsi ini dapat selesai dengan baik.

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

Penulis menyadari bahwa selesainya skripsi ini tidak terlepas dari bantuan, bimbingan, dorongan dan doa dari banyak pihak, sejak masa perkuliahan hingga selesainya penyusunan skripsi ini. Untuk itu, ucapan terimakasih penulis haturkan kepada kepada semua pihak yang membantu dalam pembantu dalam penyelesaian skripsi ini.

Depok, 23 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 : Harris Setyo Fernanto  
NPM : 0606068240  
Program Studi : S1 Instrumentasi Elektronika  
Departemen : Fisika  
Fakultas : Matematika dan Ilmu pengetahuan Alam  
Jenis karya : Skripsi

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

**SEISMIK DATA LOGGER TERSINKRONISASI WAKTU GPS  
BERBASIS MIKROKONTROLLER H8/3069 F**

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 : 27 Juni 2011

Yang menyatakan

( Harris Setyo Fernanto)

Nama : Harris Setyo Fernanto  
Npm : 0606068240  
Judul : Seismik Data Logger Tersinkronisasi waktu GPS  
Berbasis Mikrokontroler H8/3069F

## ABSTRAK

Sistem logger data gelombang seismik merupakan sistem pengambilan, dan pengumpulan, pemrosesan data getaran yang terjadi di permukaan bumi dengan metode penjalaran gelombang akustik/elastik dari bawah permukaan bumi. Dalam eksplorasi air tanah dan hidrokarbon, metode ini penting digunakan untuk mendapatkan informasi mengenai litologi dan fluida bawah permukaan bumi sehingga pada akhirnya nanti peneliti dapat menentukan keberadaan air ataupun sumber daya mineral lainnya. Sistem logger ini menggunakan mikrokontroler 16-bit H8/3069F dengan ADC (*Analog to Digital Converter*) internal disertai dengan display waktu GPS (*Global Positioning System*) pada Hardware dan PC. Pengumpulan dan pemrosesan data menggunakan GUI (*Graphical User Interface*) berbasis Python 2.5 yang merupakan program *open-source* dan *multi platform*, serta MySQL python sebagai sistem penyimpanan data, maka kebutuhan akan peralatan sistem logger data seismik dapat diperoleh dengan biaya yang lebih murah dan menjamin ketepatan data yang dihasilkan. Perangkat lunak yang bekerja pada mikrokontroler dibuat menggunakan bahasa C yang dikategorikan sebagai bahasa *mid-level* agar mudah diimplementasikan pada mikrokontroler.

Kata kunci: Logger, Seismik, 16 bit, H8/3069F, GPS, Python 2.5, MySQL, bahasa C.

Name : Harris Setyo Fernanto  
Department : Physics  
Title of essay : *Seismic Data Logger Synchronized by GPS Time Based on Microcontroller H8/3069F*

## **ABSTRACT**

*Data logger system is a system of seismic wave-making, collecting, and processing of vibration data that occurred in the earth's surface by the method of acoustic waves spreading / elastic from beneath the earth's surface. In the exploration of ground water, this method is important used to obtain information about litologi and fluid under the surface of the earth so eventually the researchers can determine the presence of water or other mineral resources. This logging system using 16-bit microcontroller with H8/3069F ADC (Analog to Digital Converter) and GPS (Global Positioning System) time on hardware and software. Collecting and processing data using a GUI (Graphical User Interface) based on Python 2.5 is an open-source programs and multi-platform, and MySQL python as a data storage system, the need for equipment seismic data logger system can be obtained with a cheaper cost and ensure the accuracy of the data produced. Software that works on the microcontroller built using C language categorized as mid-level language is easy implement in the microcontroller.*

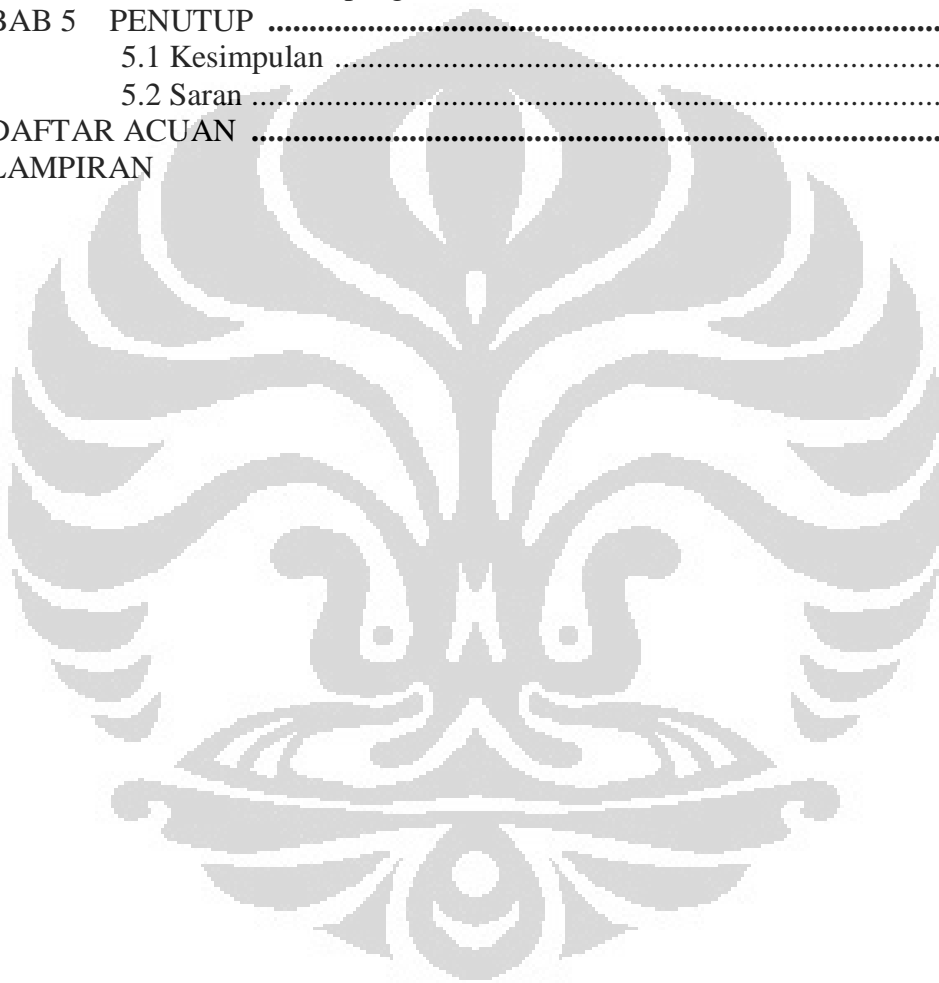
*Keywords: Logger, Seismic, 16 bit, H8/3069F, GPS (Global Positioning System), Python, database MySQL, GUI (Graphical User Interface)*

# DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	ii
KATA PENGANTAR .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
DAFTAR ISI .....	vi
DAFTAR TABEL .....	viii
DAFTAR GAMBAR .....	ix
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Batasan Masalah .....	2
1.3 Tujuan dan Manfaat Penelitian .....	3
1.4 Metodologi Penelitian .....	3
1.5 Sistematika Penulisan .....	5
BAB 2 LANDASAN TEORI .....	6
2.1 Gelombang Seismik .....	6
2.1.1 Akuisisi Gelombang Seismik .....	7
2.2 <i>Geophone</i> .....	8
2.2.1 Sensitivitas <i>Geophone</i> .....	9
2.3 Rangkaian <i>Direct Adder</i> .....	10
2.4 Penguat Instrumentasi .....	11
2.5 <i>Analog to Digital Converter</i> (ADC) .....	12
2.6 <i>Global Positioning System</i> (GPS) .....	15
2.7 <i>Real Time Clock</i> (RTC) .....	19
2.8 Komunikasi Serial .....	29
2.9 Mikrokontroler H8/3069F .....	33
2.10 Python .....	39
2.11 <i>DataBase</i> MySQL .....	40
BAB 3 PERANCANGAN ALAT .....	42
3.1 Perangkat Keras .....	42
3.1.1 Rangkaian Penguat Instrumentasi .....	43
3.1.2 Rangkaian <i>Direct Adder</i> .....	44
3.1.3 Pengkonversi Analog ke Digital Internal Mikrokontroler H8/3069F .....	44
3.1.4 Modul GPS GARMIN .....	48
3.1.5 <i>Real Time Clock</i> (RTC) .....	50
3.1.6 LCD 16x2 LMB162A .....	51
3.2 Perangkat Lunak .....	52
3.2.1 Pemrograman ADC Internal Mikrokontroler H8/3069F .....	52
3.2.2 Pemrograman Tampilan GUI ( <i>Graphical User Interface</i> ) Python .....	54
3.2.3 Penyimpanan Data .....	55
BAB 4 ANALISA HASIL PENELITIAN .....	56



4.1 Perangkat Keras .....	56
4.1.1 Uji Rangkaian Penguat Instrumentasi .....	56
4.1.2 Uji Linearitas Rangkaian ADC ( <i>Analog to Digital Converter</i> ) Internal Mikrokontroler H8/3069F.....	58
4.1.3 Uji Sistem Waktu .....	61
4.2 Perangkat Lunak .....	63
4.2.1 Pemrograman ADC Internal Mikrokontroler H8/3069F Mode <i>Scan</i> .....	63
4.2.2 Pemrograman GUI Python Sistem Akuisisi Data Gelombang . Seismik .....	64
4.3 Hasil Data Lapangan .....	65
BAB 5 PENUTUP .....	68
5.1 Kesimpulan .....	68
5.2 Saran .....	68
DAFTAR ACUAN .....	69
LAMPIRAN	



## DAFTAR TABEL

Tabel 2.1. Deskripsi pin DS1307.....	20
Tabel 2.2. Kontrol Register DS1307 .....	24
Tabel 2.3. Frekuensi Ouput Square Wave DS1307 .....	25
Tabel 2.4. Daftar Fungsi Pin DB-9.....	30
Tabel 2.5. <i>Standard Port Addresses</i> DB-9.....	32
Tabel 2.6. Daftar Rehister DB-9.....	32
Table 3.1. Konfigurasi register ADC internal mikrokontroler H8/3069F .....	46
Tabel 3.2 .Konfigurasi pin ADC internal mikrokontoler H8/3069F .....	46
Tabel 3.3. Deskripsi Pin Gramin .....	49
Tabel 3.4. Data \$GPRMC .....	50
Tabel 3.5. Posisi DIP Switch Mode-3.....	53
Tabel 3.6. Posisi DIP Switch Mode-7.....	54
Tabel 4.1. Data penguatan rangkaian penguat nstrumentasi.....	57
Tabel 4.2. Tegangan <i>input</i> dan <i>output</i> pada ADC internal mikrokontroler H8/3069F.....	59

## DAFTAR GAMBAR

Gambar 1.1	Skematik perancangan sistem .....	4
Gambar 1.2	Alur tahap penelitian.....	4
Gambar 2.1	Sinyal seismik akibat pemampatan dan peregangan partikel medium oleh sumber usikan.....	6
Gambar 2.2	<i>Seismic field work: Source, geophone spread and terminal</i> .....	7
Gambar 2.3	Geophone .....	8
Gambar 2.4	Sensor geophone dengan analisis rangkaian .....	9
Gambar 2.5	<i>Contoh Rangkaian Direct Adder</i> .....	10
Gambar 2.6	IC AD620 .....	11
Gambar 2.7	Rangkaian dasar <i>integrating converter</i> .....	13
Gambar 2.8	Rangkaian dasar SAR ADC .....	13
Gambar 2.9	Rangkaian flash ADC 3-bit.....	14
Gambar 2.10	<i>GPS Satellite Constellation</i> .....	16
Gambar 2.11	<i>GPS satellite</i> .....	16
Gambar 2.12	<i>GPS Signal Structure</i> .....	17
Gambar 2.13	<i>Autonomous Navigation</i> .....	17
Gambar 2.14	<i>Differential Phase position</i> .....	18
Gambar 2.15	<i>Differentially corrected positioning (DGPS)</i> .....	18
Gambar 2.16	Bentuk Fisik DS1307.....	20
Gambar 2.17	Rangkaian Operasi DS1307 .....	21
Gambar 2.18	Blok Diagram DS1307.....	21
Gambar 2.19	Address Map DS1307.....	23
Gambar 2.20	DS1307 Timekeeper Registers.....	24
Gambar 2.21	Tipikal konfigurasi bus i2c.....	25
Gambar 2.22	Transfer Data Pada i2c Serial Bus .....	27
Gambar 2.23	Data Write – Slave Receiver Mode .....	28

Gambar 2.24	Data Read – Slave Transmitter Mode.....	28
Gambar 2.25	Urutan Letak Pin Female RS232 DB-9 .....	31
Gambar 2.26	Urutan Letak Pin Male RS232 DB9 .....	31
Gambar 2.27	Perkembangan Mikrokontroler H8.....	33
Gambar 2.28	Daftar perkembangan mikrokontroler yang menggunakan mikroprocessor <i>H8/300H</i> .....	34
Gambar 2.29	Fitur-fitur dari mikrokontroler <i>H8/3069F</i> .....	35
Gambar 2.30	Memori map mode single-chip.....	38
Gambar 3.1	Blok diagram sistem akuisisi data seismik tersinkronisasi waktu GPS .....	43
Gambar 3.2	Konfigurasi pin IC AD620.....	43
Gambar 3.3	Blok diagram ADC internal mikrokontroler H8/3069F.....	45
Gambar 3.4	<i>Read only register</i> 16 bit.....	46
Gambar 3.5	<i>A/D control register</i> (ADCR).....	48
Gambar 3.6	<i>GPS GARMIN</i> .....	48
Gambar 3.7	Interkoneksi Serial Port Komputer .....	49
Gambar 3.8	Skematik rangkaian RTC .....	51
Gambar 3.9	LCD LMB162A.....	51
Gambar 3.10	Alur operasi ADC mode scan.....	53
Gambar 3.11	Langkah-langkah <i>writing</i> program ADC Internal .....	54
Gambar 3.12	Flowchart tampilan GUI Phyton .....	55
Gambar 4.1	Grafik Input Vs Output Rangkaian Penguat Instrumentasi.....	57
Gambar 4.2	Input dan output rangkaian penguatan instrumentasi .....	58
Gambar 4.3	Grafik output ADC internal mikrokontroler H8/3069.....	60
Gambar 4.4	RTC dan Hasil Output pada hyperterminal PC .....	61
Gambar 4.5	Data waktu dan tanggal GPS setelah parsing .....	62
Gambar 4.6	Display Waktu di LCD .....	63
Gambar 4.7	Hasil konversi ADC pada hyperterminal dengan waktu GPS.....	63
Gambar 4.8	GUI Seismik Data Logger.....	64
Gambar 4.9	Database Seismik Data Logger .....	65
Gambar 4.10	Uji Sensor Geopon.....	65
Gambar 4.11	Grafik Data Seismik.....	66

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Mikrokontroler 16 bit H8/3069F telah diaplikasikan pada akuisisi data seismik untuk eksplorasi air tanah (Kurnadi 2009). Akan tetapi aplikasi tersebut belum didukung oleh sistem pewaktuan dengan menggunakan sinyal GPS (*Global Positioning System*) dan RTC (*Real Time Clock*). Berdasarkan hal itulah, maka penelitian ini dilakukan.

Seismik data logger merupakan salah satu proses yang sangat diperlukan dalam suatu penelitian. Dengan memanfaatkan sensor geophone sebagai pendeteksi gelombang seismik yang ada di bumi, maka proses logger data bisa berjalan. Geophone adalah sensor yang berfungsi mengubah getaran mekanik menjadi sinyal listrik (someone 1987). Sensor ini biasa digunakan pada proses akuisisi gelombang seismik didaerah yang diduga mengandung hidrokarbon seperti minyak dan gas bumi. Pada proses logger data ini, terlebih dahulu kita harus mengetahui besaran elektrik yang terdapat di bumi untuk mengetahui interpretasi dan fenomena yang terjadi di bumi tersebut. Data gelombang seismik tersebut kemudian diproses oleh mikrokontroler yang bertugas sebagai logger data dan sistem kontrol. Agar sistem logger data memiliki tingkat validitas yang tinggi dan kesalahan (error) yang minimal, maka diperlukan adanya koneksi dengan sinyal GPS dan RTC dimana akan dihasilkan informasi waktu secara real time.

GPS adalah suatu satelit berbasis sistem navigasi yang menyediakan informasi posisi dan waktu secara kontinyu. Informasi yang diberikan oleh GPS bersifat global dan resistif terhadap kondisi cuaca apapun. Pada awalnya GPS dikembangkan secara penuh hanya untuk keperluan militer, namun karena tingkat kemanfaatannya yang besar dan terutama untuk sebuah alasan keamanan, akhirnya GPS bisa digunakan oleh rakyat sipil. RTC berfungsi sebagai kalender dan jam elektronik di mana perhitungan hari, tanggal, bulan, tahun, jam, menit dan detik tersimpan di memori dengan alamat-alamat

tertentu, plus alarm yang dapat diprogram keaktifannya. Sistem perhitungan waktu dan kalender pada RTC berjalan secara kontinyu, dan secara otomatis akan dilakukan sinkronisasi waktu dengan GPS untuk koreksi, RTC dilengkapi dengan battery sehingga walaupun power supply utama dimatikan akan bekerja secara kontinyu sesuai berjalannya waktu. RTC pada system ini menggunakan IC DS1307 juga merupakan rangkaian untuk *timestamp* pada event/data yang masuk ke data logger. RTC dalam rangkaian ini menggunakan crystal clock 32,768kHz yang terhubung dengan pin 1 dan pin 2 DS1307, dan serial clock (SCL) pin 6, dan data line (SDA) pin 5 masing-masing di *pull up* dengan resistor 10k ke VCC serta dihubungkan dengan Port mikrokontroler secara berturut-turut.

Sistem seismik data logger ini menggunakan mikrokontroler 16-bit H8/3069f yang di dalamnya terdapat *microprocessor* H8/300h. Digunakannya mikrokontroler keluarga H8 dimaksudkan untuk mempermudah perpindahan pembelajaran dari mikrokontroler 8-bit ke 16-bit. Selain itu, mikrokontroler H8/3069 F juga memiliki periferil dan fitur yang lebih banyak serta konsumsi power yang lebih rendah [2]. Bahasa dan kompilator yang digunakan adalah bahasa C dan kompilator motorola yang membuat file format ".mot".

Dengan memanfaatkan alat ini maka kebutuhan akan data seismik menjadi lebih tepat dan akurat. Selain itu, sistem logger data ini juga menggunakan GUI (*Graphic User Interface*) berbasis Python yang *open-source* dan *multiplatform* sehingga menjadikan sistem ini murah dan dapat dijalankan pada beberapa *platform/operating system* komputer. Sistem ini menggunakan database MySQL yang berguna dalam penyimpanan dan analisa data.

## 1.2 Batasan Masalah

Pada perancangan alat ini hanya dibatasi pada *logger data dan display waktu* di hardware dan PC. Pada perancangan perangkat keras digunakan penguat instrumentasi, rangkaian *direct adder*, pengkonversi tegangan analog ke digital (*Analog to Digital Converter*) internal mikrokontroler H8/3068 F, sinyal GPS (*Global Positioning System*) sebagai input waktu, RTC (*Real Time*

*Clock*) sebagai kontrol output dan display waktu serta mikrokontroler H8/3068 F sebagai pusat kontrol dan pengolahan data. Sedangkan pada perangkat lunaknya digunakan GUI (*Graphic User Interface*) yang menggunakan bahasa pemrograman python sebagai media komunikasi antara pengguna dengan sistem dan database MySQL sebagai penyimpan data.

### 1.3 Tujuan dan Manfaat Penelitian

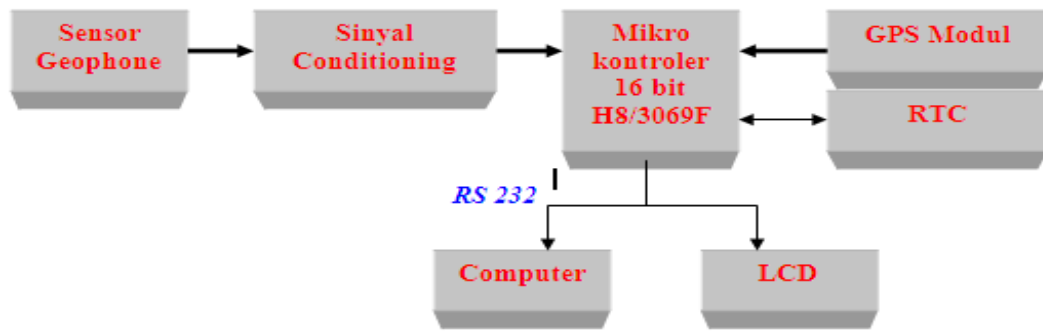
1. Mempelajari dan memahami logger data seismik dengan mikrokontroler H8/3069 F.
2. Merancang sistem logger data gelombang seismik yang tersinkronisasi waktu GPS dengan menggunakan Mikrokontroler H8/3069 F dan GUI berbasis Python.

### 1.4 Metodologi Penelitian

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

- a. Studi Peralatan  
Studi peralatan digunakan penulis untuk mempelajari karakteristik dan spesifikasi alat yang akan digunakan sehingga penulis mendapatkan pembelajaran yang tepat tentang alat yang akan dipakai.
- b. Studi Literatur  
Studi literatur dilakukan untuk memperoleh informasi, landasan teori, dan pustaka yang berkaitan dengan masalah ini diperoleh dari literatur, penjelasan yang diberikan dosen pembimbing, rekan-rekan mahasiswa, internet dan buku-buku yang berhubungan dengan tugas akhir penulis.
- c. Penelitian Laboratorium  
Penelitian laboratorium dilakukan untuk merakit, membuat alat, dan menganalisis alat yang telah dibuat.
- d. Pengambilan Data  
Data yang diambil adalah data seismoelektrik yang terdapat di lapangan.

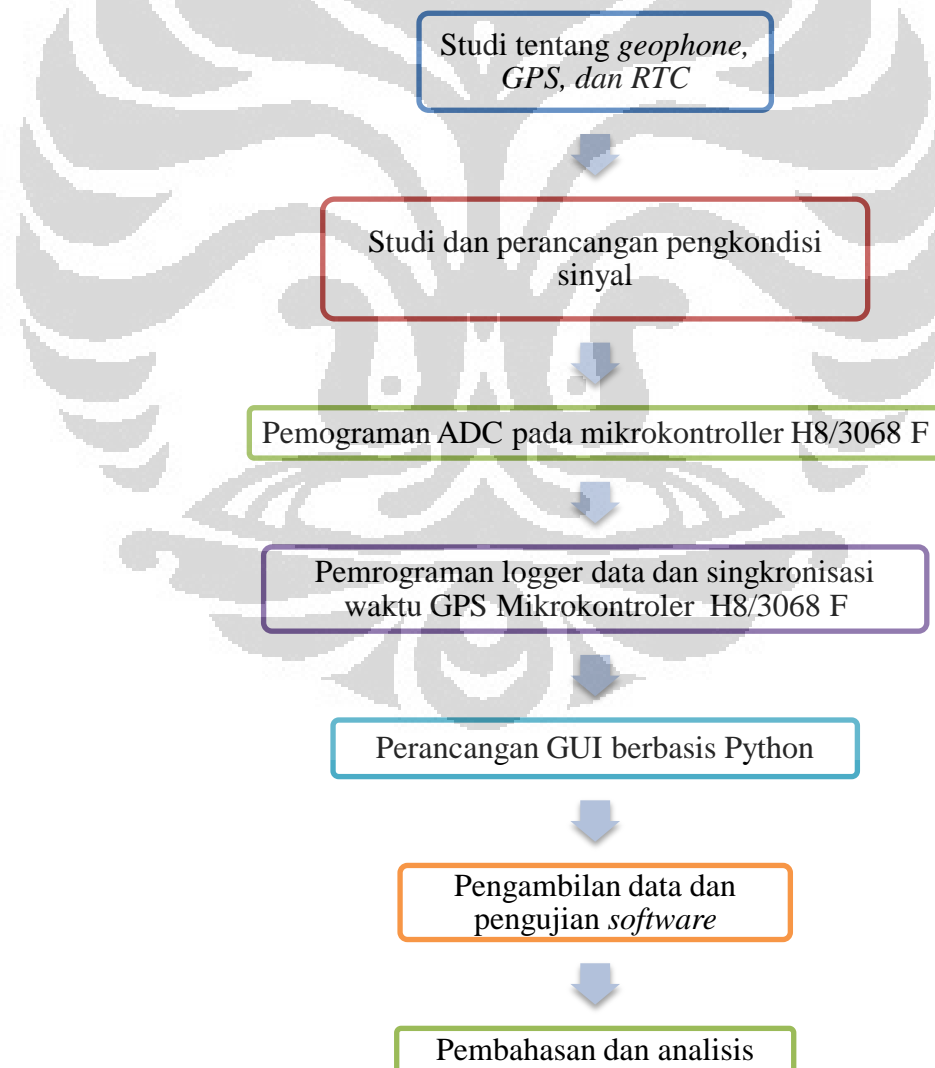
Skematik perancangan sistem logger data gelombang seismik :



Gambar 1.1.

Skematik perancangan sistem

Adapun alur tahapan pada perancangan sistem logger data gelombang seismik ini adalah :



Gambar 1.2 Alur tahap 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 penggunaan perangkat keras dan perangkat lunak yang mendukung aplikasi Python pada sistem logger data gelombang seismik.

4. Bab 4 Analisis Hasil Penelitian

Bab ini berisi penjelasan mengenai hasil penelitian dan juga analisis dari sistem logger data gelombang seismik 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.

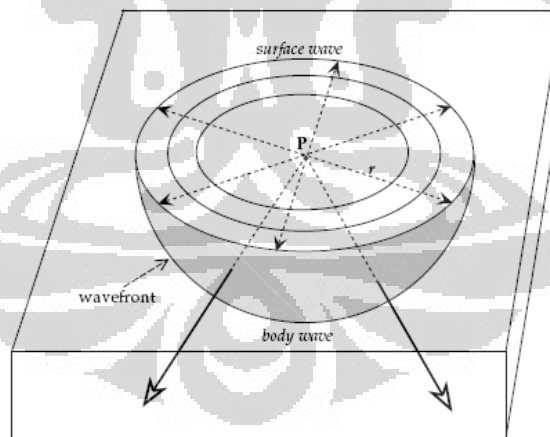
## BAB 2

### LANDASAN TEORI

#### 2.1 Gelombang Seismik

Gelombang seismik adalah gelombang mekanis yang muncul akibat adanya getaran pada permukaan bumi. Perambatan gelombang seismik adalah perambatan gangguan pada medium elastis sehingga gelombang seismik disebut juga gelombang elastis karena osilasi partikel-partikel medium terjadi akibat interaksi antara gaya gangguan melawan gaya-gaya elastis. Interaksi ini dapat menimbulkan gelombang longitudinal, gelombang transversal dan kombinasi antara keduanya (Lowrie, 2007).

Pada dasarnya gelombang seismik juga merupakan suatu bentuk perambatan energi yang dalam perambatannya mengalami suatu gangguan. Gelombang ini merupakan gelombang suara yang merambat dalam medium elastis. Pada proses ini energinya dapat berubah bentuk menjadi energi gerak dan dapat dipantulkan ke tempat semula.



Gambar 2.1. Sinyal seismik akibat pemampatan dan peregangan partikel medium oleh sumber usikan

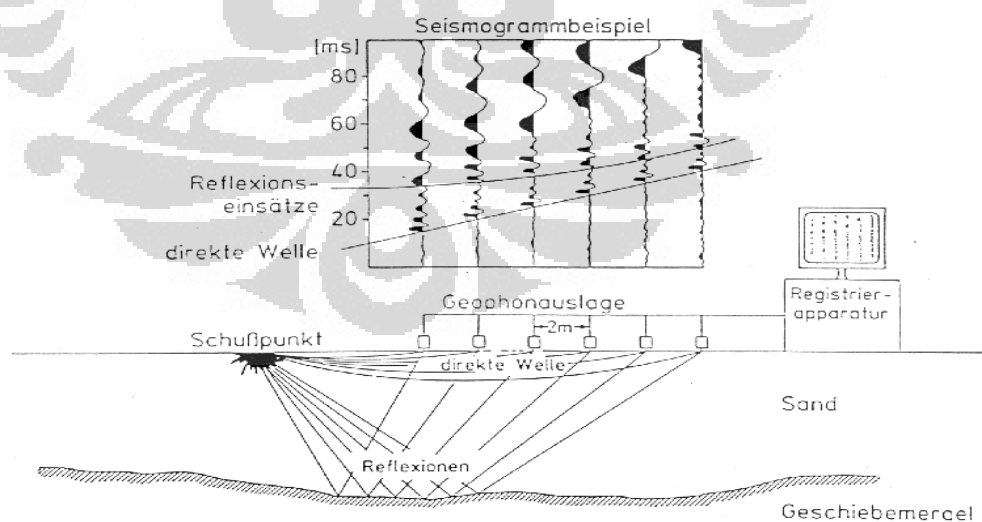
Sumber gelombang seismik pada mulanya berasal dari gempa bumi alam yang dapat berupa gempa vulkanik maupun gempa tektonik. Akan tetapi dalam seismik eksplorasi sumber gelombang yang digunakan adalah gelombang seismik buatan. Ada beberapa macam sumber gelombang seismik buatan, seperti : dinamit, benda jatuh, *air gun*, *water gun*, *vaporchoc*, *sparker*, maupun *vibroseis*.

Sumber gelombang seismik buatan tersebut pada hakekatnya membangkitkan gangguan sesaat dan lokal yang disebut sebagai gradien tegangan (stress). Gradien tegangan mengakibatkan terganggunya keseimbangan gaya-gaya di dalam medium sehingga terjadi pergeseran titik materi yang menyebabkan deformasi yang menjalar dari suatu titik ke titik lain. Deformasi ini dapat berupa pemampatan dan perenggangan partikel-partikel medium yang menyebabkan osilasi densitas/tekanan maupun pemutaran/rotasi partikel-partikel medium. Apabila medium bersifat elastis sempurna maka setelah mengalami deformasi sesaat tadi medium kembali ke keadaan semula.

### 2.1.2 Akuisisi Gelombang Seismik

Tujuan pokok dari penelitian gelombang seismik adalah untuk merekam gerakan tanah yang disebabkan oleh sumber di dalam bumi. Rekaman tersebut dalam bentuk dengan waktu. Selain itu sistem data akuisisi gelombang seismik ditujukan untuk :

- Menghasilkanlah suatu pulsa seismic dengan suatu sumber
- Mendeteksi gelombang seismik di dalam tanah dengan suatu transducer.
- Merekam dan gelombang seismik dari seismograph.



Gambar 2.2 *Seismic field work: Source, geophone spread and terminal*

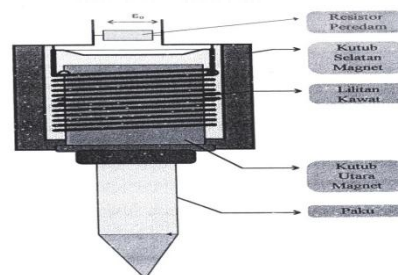
## 2.2 Geophone

Geophone merupakan sebuah alat yang sangat penting dalam eksplorasi seismik. Alat ini berfungsi untuk mendeteksi dan merekam gelombang seismik yang ada pada permukaan tanah. Layaknya sebuah sensor yang merubah besaran fisika menjadi besaran fisika lainnya khususnya besaran listrik, geophone akan mengkonversi pergerakan tanah menjadi tegangan. Geophone banyak digunakan pada banyak aplikasi seperti untuk gempa bumi, eksplorasi minyak, pertambangan dan sumber daya alam lainnya.

Geophone umumnya digunakan di atas permukaan tanah. Alat ini bekerja dengan prinsip induksi elektromagnetik. Pada umumnya geophone terdiri atas dua buah komponen utama yaitu kumparan dan magnet permanen. Salah satu dari komponen tersebut diletakkan menyatu dengan permukaan bumi sedemikian rupa sehingga akan mengikuti getaran bumi bila ada gelombang seismik yang menjalar di permukaan. Sedangkan komponen lainnya tergantung pada pegas dan dianggap tidak bergerak ketika ada gelombang seismik yang datang. Ketika permukaan tanah bergerak vertikal, magnet akan bergerak relatif terhadap koil/kumparan, Gerakan relatif antara komponen yang diam dan bergerak akibat dari gerakan vertikal bumi akan menimbulkan tegangan induksi diantara terminal koil.

Tegangan yang dihasilkan oleh geophone sebanding dengan:

- Kuatnya medan magnet yang dihasilkan oleh magnet permanen,
- Jumlah lilitan kawat,
- Diameter kumparan,
- Kecepatan gerak kumparan terhadap magnet.



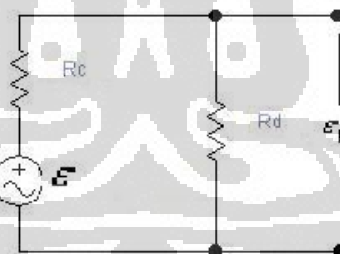
Gambar 2.3. Geophone

Tegangan induksi yang terjadi pada geophone ternyata mampu membawa informasi mengenai keadaan lapisan tanah pada permukaan bumi. Kumputan geophone dan pegas merupakan sebuah sistem osilator dengan frekuensi alami antara 4-15 Hz untuk refleksi dan 1-10 Hz untuk refraksi. Kumputan akan terus beresonansi setelah tidak ada getaran, maka diperlukan sebuah gerakan untuk meredamnya. Hal ini didapatkan pada bagian lilitan kawat di sebuah logam (Telford, 1976).

### 2.2.2 Sensitivitas Geophone

Seperti yang telah dibahas sebelumnya bahwa kecepatan getaran dari lilitan kawat terhadap magnet akan menimbulkan tegangan induksi. Disini terjadi transfer energi mekanik menjadi energi listrik. Dalam perumusannya ditulis sebagai berikut:

$$\varepsilon = Bl \frac{dz}{dt} \quad (2.18)$$



Gambar 2.4. Sensor geophone dengan analisis rangkaian

Dengan menggunakan analisis secara rangkaian elektronika dengan menambah resistor peredam ( $R_d$ ) maka tegangan keluaran menjadi :

$$\varepsilon_0 = \left( \frac{R_d}{R_d + R_c} \right) \varepsilon \quad (2.19)$$

Dengan  $R_c$  adalah nilai hambatan yang terkandung pada lilitan. Tegangan yang dihasilkan per satuan kecepatan disebut sensitivitas intrinsik efektif dari geophone yang dilambangkan dengan  $G$ . Maka diperoleh sensitivitas intrinsik efektif yaitu:

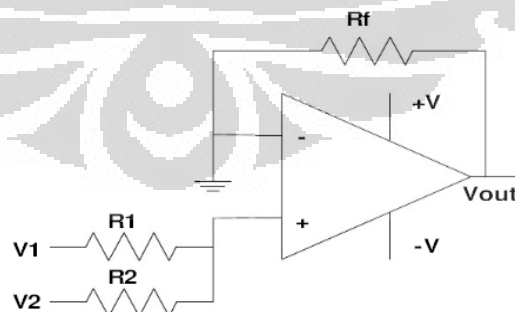
$$G_{\varepsilon} = \frac{\varepsilon_0}{\left(\frac{dz}{dt}\right)} = \left(\frac{dt}{dz} \frac{R_d}{R_d + R_c}\right) \varepsilon \quad (2.20)$$

Dalam bentuk yang lebih lengkap dapat dilihat hubungan antara tegangan induksi dengan kecepatan getaran tanah dapat ditulis kembali sebagai :

$$\varepsilon_0 = \left(\frac{R_d}{R_d + R_c}\right) Bl \frac{dz}{dt} \quad (2.21)$$

### 2.3 Rangkaian *Direct Adder*

Rangkaian *Direct Adder* (Penjumlah Langsung) adalah rangkaian yang outputnya merupakan hasil jumlah dari tegangan-tegangan input. Rangkaian ini merupakan hal khusus dari rangkaian Penjumlah-Pengurang dengan membuat semua input masuk ke terminal non-inverting, sedangkan terminal inverting digroundkan.



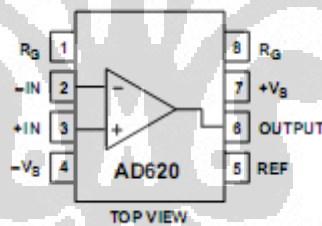
Gambar 2.5 Contoh Rangkaian *Direct Adder*

## 2.4 Penguat Instrumentasi (*Instrumentation Amplifier*)

Dalam perancangan suatu sistem pengukuran ataupun akuisisi data terkadang diperlukan suatu rangkaian penguat yang berfungsi menguatkan sinyal-sinyal rendah seperti sinyal tubuh ataupun sinyal seismik. Salah satu rangkaian penguat yang dapat digunakan adalah penguat instrumentasi (*instrumentation amplifier*) atau yang disebut juga penguat *transducer* atau penguat beda.

Penguat instrumentasi adalah rangkaian op-amp yang digunakan untuk memperkuat signal dari transduser, umumnya sinyalnya sangat rendah didalam sinyal *common* yang tinggi. Oleh karena itu, penguat instrumentasi harus memiliki penguat yang besar, offset yang rendah, dan CMMR yang tinggi.

Pada dasarnya, penguat instrumentasi terdiri dari tiga buah penguat operasional (*operational amplifier/op-amp*). Dua buah *op-amp* pertama sebagai *buffer* sedangkan *op-amp* ketiga sebagai penguat beda (*differensial amplifier*). Namun, ada juga yang menggunakan satu buah op-amp yaitu IC AD620. Seperti ditunjukkan pada gambar 2.6 dibawah.



Gambar 2.6 IC AD620

Penguat intrumentasi memiliki semua sifat dari *op-amp*, yaitu :

1. Penguatan besar
2. Impedansi input tinggi
3. Impedansi output rendah
4. Tegangan offset rendah
5. CMRR tinggi

## 2.5 Analog to Digital Converter (ADC)

Pada umumnya besaran di alam adalah besaran analog yang memiliki ciri bahwa perubahan nilai dari besaran tersebut bersifat *smooth*, sedangkan besaran digital nilainya berjenjang yaitu ada satu nilai perubahan terendah.

Suatu sinyal analog jika diolah dengan pengolah analog hasilnya juga akan analog, sedangkan sinyal digital jika diolah oleh pengolah digital maka hasilnya juga akan digital. Pemrosesan secara digital seringkali diperlukan karena memiliki keuntungan dibandingkan dengan pemrosesan secara analog, diantaranya adalah :

- Lebih kebal terhadap *noise*,
- Lebih mudah diproses,
- Lebih mudah diperbaiki, dan
- Lebih murah

Contoh sederhana dari gejala digital adalah berjalan menaiki tangga, sedangkan gejala analog adalah berjalan pada undakan.

Rangkaian ADC (*Analog to Digital Converter*) merupakan rangkaian yang berfungsi untuk mengubah sinyal/besaran analog menjadi sinyal digital atau yang lebih dikenal dengan bilangan biner. Umumnya, bilangan biner tersebut kemudian dikelompokkan lagi kedalam bit. Sebagai contoh resolusi sebuah ADC 8 bit, maka ADC tersebut dapat mengkodekan nilai input sebanyak  $2^8 = 256$  level sinyal input analog yang berbeda.

Terdapat beberapa metode untuk mengkonversi sinyal analog menjadi digital, diantaranya adalah :

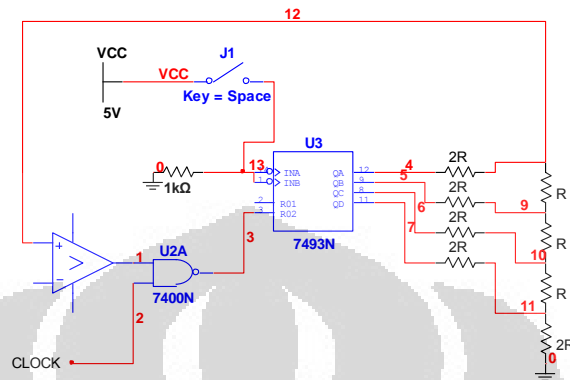
### 1. *Integrating Converter*

ADC pada tipe ini menggunakan komparator digital, *counter*, dan DAC (*Digital to Analog Converter*). Nilai input dibandingkan dengan output dari DAC, jika kedua nilai tersebut belum sama maka sinyal *clock* tetap bekerja, sebaliknya jika sudah sama/lebih maka proses perhitungan dihentikan.

Rangkaian dari ADC tipe ini ditunjukkan pada gambar (...). Komparator yang dapat digunakan adalah LM339, sedangkan *counter*-nya adalah 7493, dan DAC-nya adalah rangkaian R-2R.



Pada contoh gambar rangkaian dibawah, tegangan acuan yang digunakan adalah 5 volt, namun jika dikehendaki tegangan lainnya diperlukan *buffer open collector* diantara *counter* dengan rangkaian DAC.

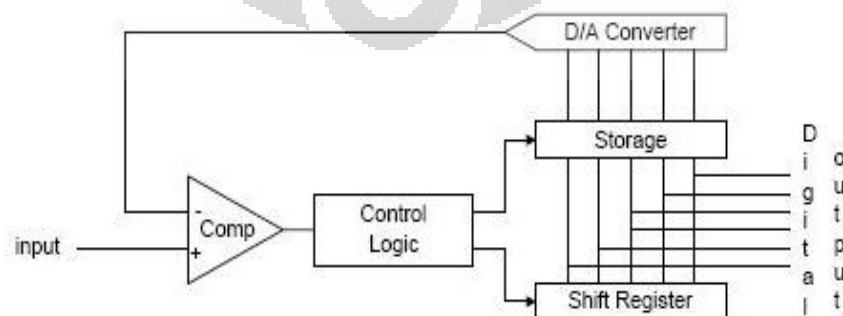


Gambar 2.7 Rangkaian dasar *integrating converter*

Rangkaian ADC ini termasuk lambat dan cara mempercepatnya misalnya menggunakan *dual slope integrating* ADC atau menggunakan SAR ADC.

## 2. Successive Approximation Register Analog to Digital Converter (SAR ADC)

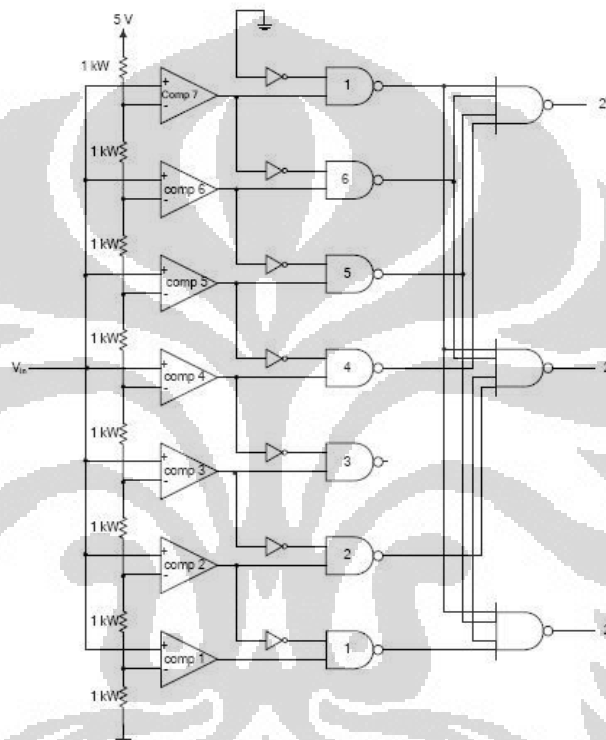
ADC ini juga menggunakan komparator digital, namun menggunakan teknik yang berbeda dengan metode *integrating*. Teknik perbandingannya adalah pertama-tama sinyal input akan dibandingkan dengan sinyal digital jika MSB (*Most Signifikan Bit*) diset (logika 1). Apabila inputnya lebih kecil, MSB=0 dan geser bit pembanding ke kanan dan lakukan pembandingan lagi. Sebaliknya, jika sinyal input lebih besar maka MSB tetap berharga 1 dan set nilai bit sebelah kanannya menjadi 1, demikian seterusnya hingga hasil konversi sesuai.



Gambar 2.8 Rangkaian dasar SAR ADC

### 3. Flash ADC

Pada ADC jenis ini, sinyal input analog langsung dibandingkan dengan beberapa komparator digital. Hasil pembandingannya kemudian dikodekan menjadi format digital. ADC jenis ini merupakan ADC tercepat. Kelemahan ADC ini adalah perlu banyak sekali komparator, untuk  $N$  bit perlu  $2^N - 1$ , sehingga biaya produksinya lebih tinggi dibandingkan metode lainnya.



Gambar 2.9 Rangkaian flash ADC 3-bit

### 2.6 Global Positioning System (GPS)

Global Positioning System (GPS) adalah satu-satunya Global Sistem Satelit Ilmu Pelayaran yang fungsional (GNSS). GPS menggunakan suatu peta bintang. Bumi mengorbitkanlah satelit yang memancarkan isyarat gelombang mikro yang tepat dan memungkinkan GPS penerima untuk menentukan penempatan mereka, kecepatan, arah, dan waktu. GPS dikembangkan oleh Departemen Keamanan Amerika Serikat. Pejabatnya menyebut adalah NAVSTAR-GPS. GPS peta bintang satelit diatur oleh Angkatan udara Amerika

Serikat Space. Lalu system navigasi satelit yang sama diikuti oleh Rusia yang bernama GLONASS, COMPASS sistem ilmu pelayaran diusulkan Negeri China, dan IRNSS India. GPS telah digunakan secara luas menggunakan alat bantu navigasi di seluruh dunia, dan suatu alat bermanfaat untuk map-making, mendaratkan kapal, mensurvei, perdagangan, penggunaan dalam bidang ilmiah, dan lain-lain. GPS juga menyediakan suatu acuan waktu tepat digunakan di dalam banyak aplikasi yang mencakup studi gempa bumi yang ilmiah, dan sinkronisasi jaringan telekomunikasi.

Suatu GPS penerima dan mengkalkulasi posisinya dengan secara hati-hati dengan pemilihan waktu isyarat yang dikirim oleh peta bintang GPS satelit yang tinggi di atas bumi. Masing-Masing satelit secara terus menerus memancarkan pesan yang berisi pesan dengan suatu garis edar tepat untuk satelit yang mengirimkan pesan (ephemeris). Isyarat ini berada di kecepatan cahaya melalui angkasa luar, dan kecepatan yang sedikit lebih lambat melalui atmosfer itu. Penerima menggunakan waktu tiba dari tiap pesan untuk mengukur jarak untuk masing-masing satelit. Lalu menentukan posisi penerima yang menggunakan posisi sudut dan ilmu ukur. Dan mengkonversi ke format yang lebih mudah dioperasikan seperti garis bujur dan garis lintang, atau penempatan pada suatu peta agar lebih mudah dilihat oleh pengguna.

GPS dapat menggunakan 3 satelit untuk memecahkan suatu posisi sehingga posisinya dapat berbentuk 3 dimensi. Penggunaan GPS dengan 3 satelit memerlukan waktu kira-kira dalam nanosecond. GPS juga dapat menggunakan 4 untuk memecahkan untuk waktu seperti halnya posisi dalam peta bumi, menentukan suatu waktu dunia yang akurat. Dengan Kata Lain, penerima menggunakan empat pengukuran untuk memecahkan untuk empat variabel:  $x$ ,  $y$ ,  $z$ , dan  $t$ .

Dibagian tempat di bumi terdiri dari 24 satelit yang mengorbit di bumi dan kira-kira jaraknya 20200 km setiap 12 jam. Setiap waktu sekitar 26 satelit yang beroperasi mengelilingi bumi.



Gambar 2.10 *GPS Satellite Constellation*

Diperlukan minimum 4 satelit yang terlihat untuk membentuk sudut cut-off  $15^\circ$  cut-off di setiap titiknya.



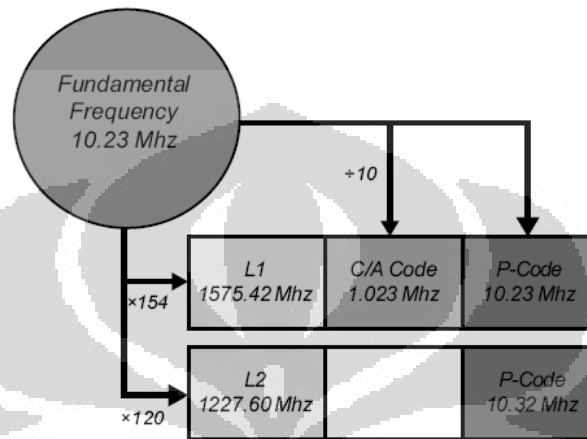
Gambar 2.11 *GPS satellite*

Beberapa satelit GPS mempunyai beberapa *atomic clock* yang sangat akurat untuk mengoperasikan frekuensi fundamental sekitar 10.23 MHz. Frekuensi ini digunakan untuk membangkitkan sinyal broadcast (siaran radio) dari satelit. Satelit broadcast membawa 2 gelombang, yaitu L-band (digunakan untuk radio) dan perjalanan ke bumi yang kecepatannya hamper sama dengan kecepatan cahaya. Frekuensi fundamental ini dibagi lagi menjadi :

1. Pembawa L1 di frekuensi 1575.42 MHz ( $10.23 \times 154$ ). Pada pembawa L1 mempunyai dua kode modulasi, yaitu kode C/A atau Coarse / Acquisition yang dimodulasikan pada frekuensi 1.023 MHz ( $10.23/10$ ) dan kode P atau Precision Code yang dimodulasikan pada frekuensi (10.23 MHz)

2. Pembawa L2 di frekuensi 1227.60 MHz ( $10.23 \times 120$ ). Pada L2 hanya satu kode yang dimodulasikan, yaitu kode P yang dimodulasikan pada frekuensi 10.23 MHz.

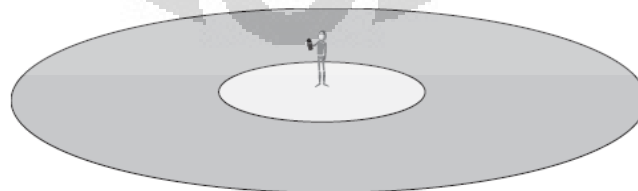
Penerima GPS menggunakan kode yang berbeda untuk membedakan antara satelit satu dengan lainnya. Kode juga dapat digunakan untuk menghitung posisi.



Gambar 2.12 *GPS Signal Structure*

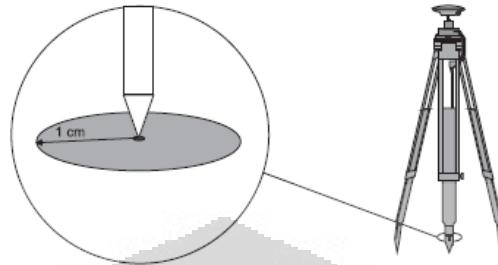
Ada beberapa metode untuk menentukan posisi dengan menggunakan GPS. Penggunaan Metode ini tergantung pada tingkat keakuratan yang diinginkan oleh pengguna dan tipe GPS yang digunakan. Adapun metode tersebut adalah :

- Autonomous Navigation* , yaitu metode GPS yang menggunakan penerima yang berdiri sendiri. Biasanya metode ini digunakan oleh kapal militer atau pejalan kaki. Keakuratan posisi ini sekitar 100 m untuk penduduk sipil dan sekitar 20 m untuk militer.



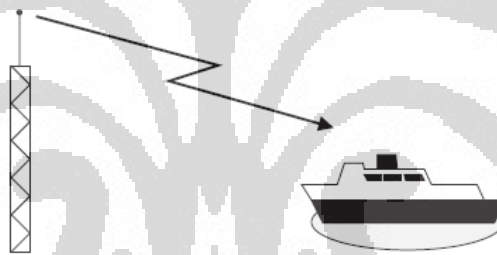
### 2.13 *Autonomous Navigation*

- b. *Differential Phase position*. Metode ini memberikan tingkat keakuratan 0.5 sampai dengan 20 mm. Metode ini digunakan oleh untuk pengontrol mesin atau pengukuran tanah.



Gambar 2.14 *Differential Phase position*

- c. *Differentially corrected positioning (DGPS)*. Metode ini mempunyai keakuratan antara 0.5 sampai dengan 5 m. Metode ini digunakan untuk navigasi kelautan, GIS data acquisition atau dalam bidang pertanian.



2.15 *Differentially corrected positioning (DGPS)*

## 2.7 Real Time Clock (RTC)

Real-Time clock (RTC) adalah jam komputer (dalam bentuk IC) yang beroperasi melacak waktu sekarang (real time). Meskipun istilah ini sering merujuk kepada perangkat di komputer pribadi, server dan embedded system, RTC hadir di hampir semua perangkat elektronik yang membutuhkan informasi waktu secara akurat.

Istilah ini digunakan untuk menghindari kebingungan dengan jam hardware biasa dimana sinyalnya hanya diatur oleh elektronika digital, dan tidak menghitung waktu dalam satuan manusia. Walaupun untuk mendapatkan

informasi waktu bisa dilakukan tanpa suatu RTC, namun dengan adanya RTC bisa didapat keuntungan :

- Konsumsi daya yang rendah
- Sistem utama bebas dari beban waktu kritis (time-critical tasks)
- Lebih akurat dari metode lain

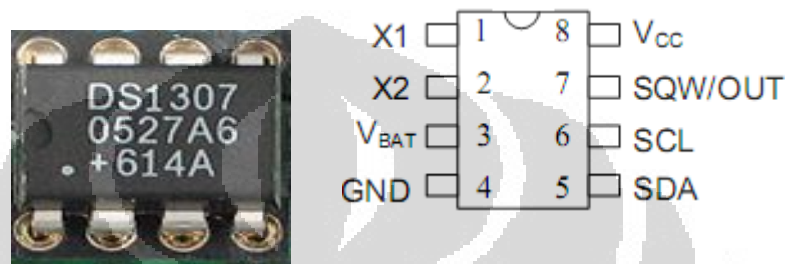
Keuntungan untuk sebuah receiver GPS adalah dapat mempersingkat waktu startup di bandingkan dengan waktu saat ini dan hasil akhir sinyal yang valid.

RTC memiliki sumber daya alternatif, sehingga mereka dapat terus menjaga informasi waktu, sementara sumber utama daya dimatikan atau tidak tersedia. Alternatif sumber daya/power biasanya baterai lithium dalam sistem lama, tapi beberapa sistem yang lebih baru menggunakan superkapasitor karena mereka dapat diisi ulang dan dapat disolder. Sumber daya alternatif juga dapat mensuplai listrik ke baterai RAM. Dalam masalah timing, RTC memanfaatkan osilator kristal, tetapi beberapa lainnya menggunakan frekuensi saluran listrik. Dalam banyak kasus, frekuensi osilator yang digunakan adalah 32,768 kHz. Ini adalah frekuensi yang sama digunakan dalam jam kuarsa dan jam tangan, dan untuk alasan yang sama, yaitu bahwa frekuensi tepat 215 siklus per detik, yang merupakan tingkat nyaman untuk digunakan dengan sederhana sirkuit counter biner

Pada sinkronisasi waktu melalui antena GPS, umumnya RTC yang digunakan agar bisa terkoneksi dengan mikrokontroler H8 adalah RTC tipe DS1307. Serial RTC DS1307 ini memiliki daya yang rendah, full binary-coded decimal (BCD) clock/calendar, dan 56 byte NV SRAM. Adress dan datanya ditransfer secara serial melalui 2 kabel bi-directional bus. Fitur jam / kalender di dalamnya menyediakan informasi detik, menit, jam, hari, tanggal, bulan, dan tahun. Jam beroperasi baik dalam format 24 jam atau 12-jam dengan indikator AM / PM. The DS1307 memiliki semacam sirkuit sensor daya yang dapat mendeteksi gangguan listrik dan secara otomatis beralih ke pasokan baterai. Bentuk fisik DS1307 ditunjukkan oleh gambar 1, sementara deskripsi pin dan tipe rangkaian

operasinya ditunjukkan oleh tabel 1 gambar 2. Spesifikasi lain dari DS1307 adalah sebagai berikut:

- Sinyal output squarewave yang bersifat “programmable”
- Mengonsumsi kurang dari 500nA pada mode baterai cadangan dengan osilator berjalan
- Beroperasi pada kisaran suhu industri  $-40^{\circ}\text{C}$  sampai  $+85^{\circ}\text{C}$

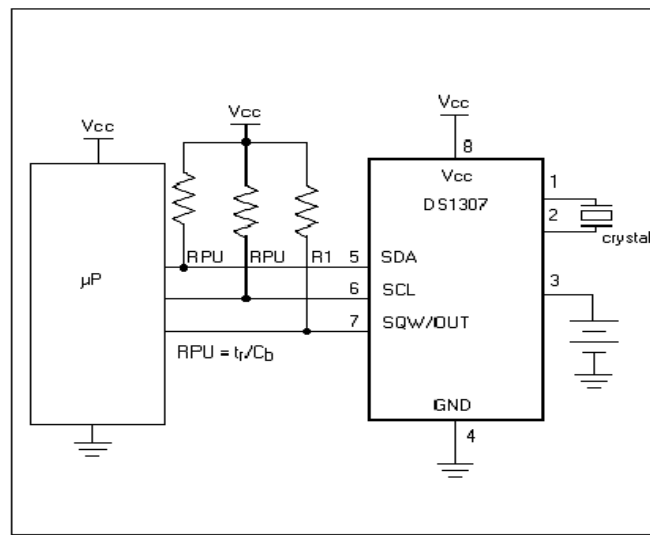


Gambar 2.16 Bentuk Fisik DS1307

Tabel 2.1 Deskripsi Pin DS1307

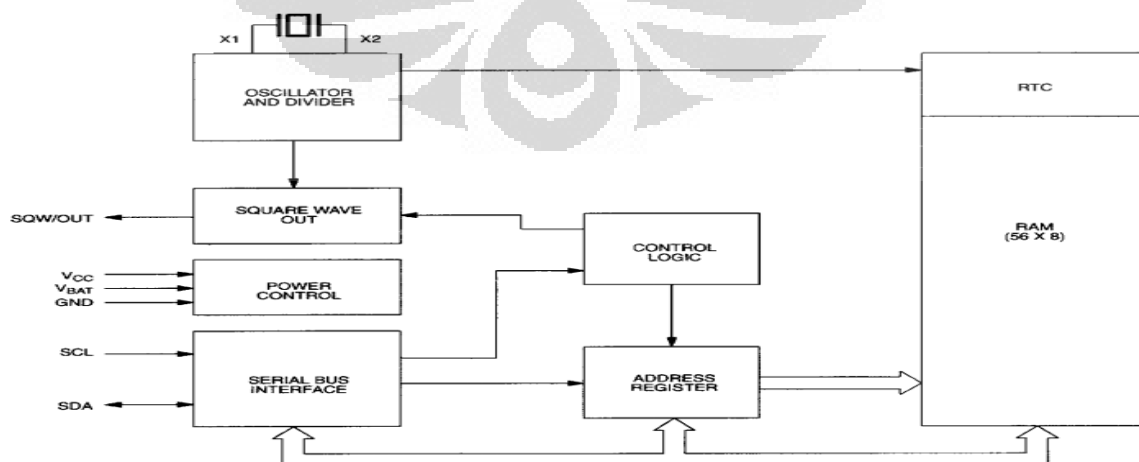
Pin	Nama	Fungsi
1,2	X1,X2	32.768kHz Crystal Connection
3	Vbat	Input baterai +3V
4	GND	Ground
5	SDA	Serial Data
6	SCL	Serial Clock
7	SQW/OUT	Square wave/Output Driver
8	Vcc	Primary Power Supply





Gambar 2.17 Rangkaian Operasi DS1307

The DS1307 beroperasi sebagai *slave devices* pada serial bus. Aksesnya diperoleh dengan mengimplementasikan kondisi START dan memberikan kode identifikasi yang diikuti dengan sebuah *register address*. Register berikutnya dapat diakses secara berurutan sampai kondisi STOP dieksekusi. Ketika  $V_{cc}$  turun di bawah  $1,25 \times V_{bat}$ , devais akan menutup sebuah akses dalam progres dan reset devais adress counter. Input ke devais tidak akan diterima saat itu untuk mencegah data error dari “penulisan” ke devais yang berada di luar sistem toleransi. Ketika  $V_{cc}$  turun di bawah  $V_{bat}$ , perangkat akan *switch* ke mode baterai cadangan arus rendah. Setelah daya meningkat, devais akan beralih dari baterai ke  $V_{cc}$  ketika  $V_{cc}$  lebih besar dari  $V_{bat} + 0.2V$  dan akan menerima input ketika  $V_{cc}$  lebih besar dari  $1,25 \times V_{bat}$ . Blok diagram pada Gambar 3 menunjukkan unsur-unsur utama dari serial RTC.



Gambar 2.18 Blok Diagram DS1307

## Signal Description

**Vcc, GND** – Daya DC disediakan pada perangkat pin ini. Vcc adalah input 5 V. Ketika 5V diterapkan dalam batas normal, perangkat sepenuhnya dapat diakses dan data dapat ditulis serta dibaca. Ketika baterai 3V tersambung ke perangkat dan Vcc di bawah  $1,25 \times V_{bat}$ , proses membaca dan menulis akan terhambat. Namun, fungsi ketepatan waktu secara kontinyu tidak akan terpengaruh oleh tegangan input yang lebih rendah. Ketika Vcc turun di bawah Vbat RAM, pencatat waktu akan beralih ke catu daya eksternal (nominal 3.0V DC) di Vbat.

**Vbat** – Input baterai untuk setiap standar sel lithium atau sumber energi lainnya adalah 3V. Tegangan baterai harus berada antara 2.0V dan 3.5V untuk beroperasi dengan baik. Sebuah baterai lithium dengan 48mAh atau lebih, akan kembali meningkatkan kinerja DS1307 selama lebih dari 10 tahun dengan tidak adanya daya pada suhu  $25^{\circ}\text{C}$ .

**SCL (Serial Clock Input)** - SCL digunakan untuk menyinkronkan perpindahan data pada serial interface.

**SDA (Serial Data Input / Output)** - SDA adalah pin I/O untuk i2c serial interface. Pin SDA dapat mengalami *open drain* dimana syaratnya adalah membutuhkan resistor pullup eksternal.

**SQW / OUT (Square Wave / Driver Output)** – Ketika di aktifkan, bit SQWE akan set ke 1, pin output SQW / OUT akan bernilai salah satu dari empat frekuensi gelombang persegi (1Hz, 4kHz, 8kHz, 32kHz). Pin SQW /OUT akan *open drain* dan membutuhkan resistor pull-up eksternal. SQW / OUT akan beroperasi dengan baik apabila Vcc atau Vbat diterapkan.

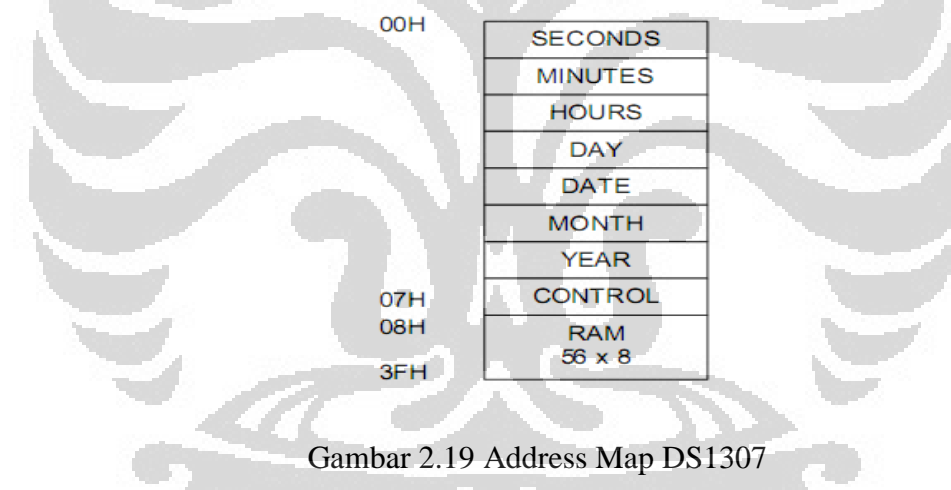
**X1, X2** - Standar untuk melakukan koneksi adalah pada frekuensi 32.768kHz crystal quartz. Sirkuit osilator internal dirancang untuk beroperasi dengan kristal yang memiliki kapasitansi muatan spesifik (CL) pada 12.5pF.

## Clock Accuracy

Akurasi jam tergantung pada akurasi kristal dan kecocokan akurasi antara muatan kapasitif rangkaian osilator dengan muatan kapasitif yang memiliki kristal terpotong. Error akan terus bertambah akibat drift frekuensi kristal yang disebabkan oleh perubahan suhu. Akumulasi noise pada sirkuit eksternal dalam sirkuit osilator dapat mengakibatkan jam berjalan cepat

## RTC AND RAM Address Map

Address map untuk RTC dan register RAM dari DS1307 ditunjukkan pada Gambar 4. Register RTC berada pada lokasi address 00h sampai 07h. Register RAM terletak di lokasi address 08h sampai 3Fh. Selama dalam akses multi-byte, ketika address pointer mencapai 3Fh, space RAM akan habis dan akan membalikannya ke lokasi 00h yang merupakan awal dari clock space.



Gambar 2.19 Address Map DS1307

## Clock And Calendar

Informasi Jam dan Kalender diperoleh dengan membaca byte register yang sesuai. Register RTC diilustrasikan pada Gambar 5. Jam dan kalender mengatur atau diawali dengan menuliskan byte register yang sesuai. Isi waktu dan register kalender dalam format BCD. Bit 7 dari register 0 adalah menghentikan jam (CH) bit. Ketika bit ini diset ke 1, osilator dinonaktifkan. Ketika clear ke 0, osilator ini diaktifkan.

	BIT7									BIT0	
00H	CH	10 SECONDS			SECONDS						00-59
	0	10 MINUTES			MINUTES						00-59
	0	12 / 24	10 HR / A/P	10 HR	HOURS						01-12 / 00-23
	0	0	0	0	0	DAY					1-7
	0	0	10 DATE		DATE						01-28/29 / 01-30 / 01-31
	0	0	0	10 MONTH	MONTH						01-12
	10 YEAR				YEAR						00-99
07H	OUT	0	0	SQWE	0	0	RS1	RS0			

Gambar 2.20 DS1307 Timekeeper Registers

**Control Registers**

Register control DS1307 digunakan untuk mengontrol operasi pin SQW / OUT

Tabel 2.2 Kontrol Register DS1307

<b>BIT 7</b>	<b>BIT 6</b>	<b>BIT 5</b>	<b>BIT 4</b>	<b>BIT 3</b>	<b>BIT 2</b>	<b>BIT 1</b>	<b>BIT 0</b>
OUT	0	0	SQWE	0	0	RS1	RS0

**OUT (Output Control):** bit ini mengontrol tingkat output dari pin SQW / OUT ketika output square wave dinonaktifkan. Jika SQWE = 0, level logika pada pin SQW / OUT adalah 1, jika 1 = OUT dan OUT 0 jika = 0.

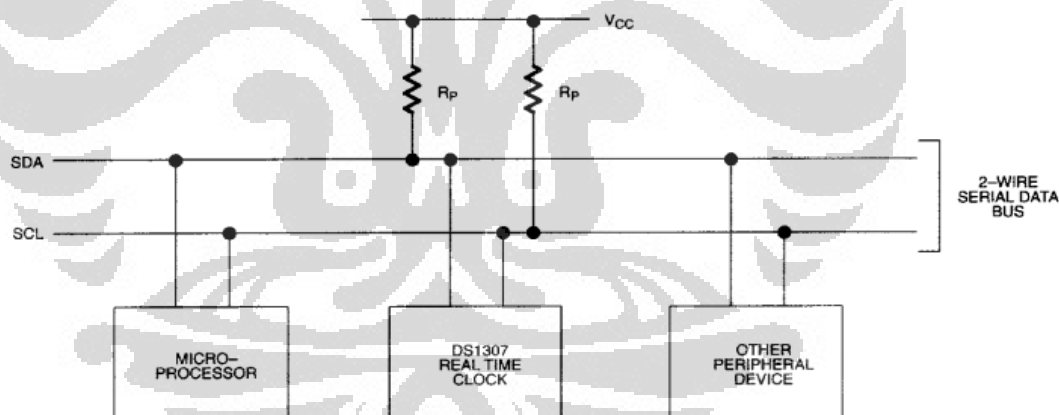
**SQWE (Square Wave Enabled):** pada bit ini, bila diset ke logika 1 maka akan mengaktifkan output osilator. Frekuensi output square wave tergantung pada nilai bit RS1 dan RS0.

**RS (Rate Select):** bit ini mengontrol frekuensi output square wave ketika output square wave telah diaktifkan. Tabel 2 mencantumkan frekuensi square wave yang dapat dipilih dengan bit RS

Tabel 2.3 Frekuensi Output Square Wave DS1307

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

Sebuah perangkat yang mengirimkan data ke bus didefinisikan sebagai transmitter dan perangkat menerima data sebagai receiver. Perangkat yang mengendalikan pesan disebut master. Perangkat yang dikendalikan oleh master disebut sebagai *slaves*. Bus harus dikontrol oleh perangkat master yang menghasilkan serial clock (SCL), mengontrol akses bus, dan menghasilkan kondisi START dan STOP. DS1307 beroperasi sebagai slave pada i2c bus. Sebuah tipikal konfigurasi bus menggunakan protokol i2c ini ditampilkan dalam Gambar 6



Gambar 2.21 Tipikal konfigurasi bus i2c

Gambar 2.22, 2.23, dan 2.24 menunjukkan dengan detail bagaimana data ditransfer melalui i2c serial bus. Ada 2 kondisi untuk transfer data :

- Transfer data dapat dimulai hanya ketika bus tidak sibuk.
- Selama transfer data, baris data harus tetap stabil setiap kali clock line bernilai HIGH. Perubahan pada data line dimana clock line bernilai HIGH akan diinterpretasikan sebagai kontrol sinyal

Sedangkan kondisi untuk bus, dapat didefinisikan seperti :

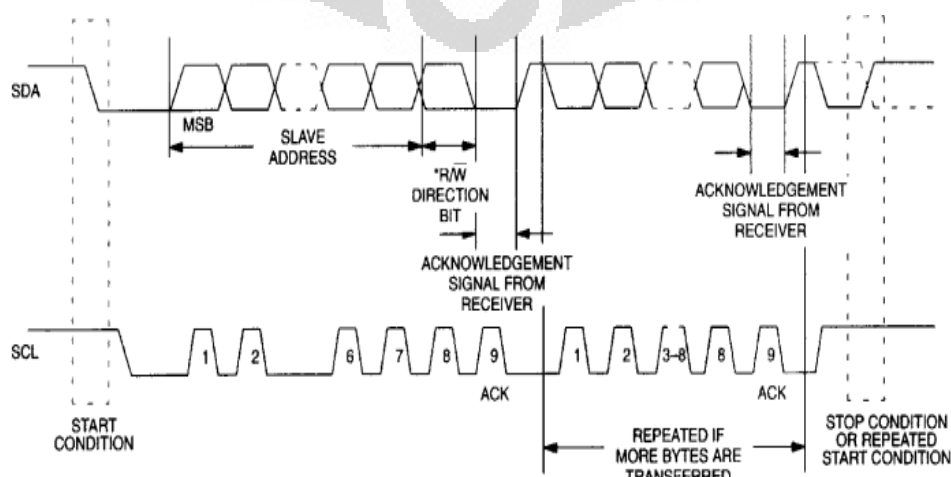
**Bus not busy** : Kedua data dan clock line tetap bernilai HIGH

**Start data transfer**: Perubahan keadaan data line dari HIGH ke LOW ketika clock bernilai HIGH, di definisikan sebagai kondisi START

**Stop data transfer** : Perubahan keadaan data line dari LOW ke HIGH ketika clock bernilai HIGH, di definisikan sebagai kondisi STOP

**Data valid**: Keadaan data line merupakan data yang valid ketika setelah kondisi START, data line stabil selama periode HIGH dari sinyal clock. Data pada line tersebut harus diubah selama periode LOW dari sinyal clock. Disana ada satu pulsa clock per bit data.

Setiap transfer data dimulai dengan kondisi START dan diakhiri dengan kondisi STOP. Jumlah byte data yang ditransfer antara kondisi START dan STOP tidak terbatas dan ditentukan oleh perangkat master. Informasi ditransfer oleh byte-wise dan masing-masing receiver akan diatur oleh bit ke-9. Dalam spesifikasi 2-wire bus, mode reguler (100kHz clock rate) dan mode fast (400kHz clock rate) dapat terdefinisi.



Gambar 2.22 Transfer Data Pada i2c Serial Bus

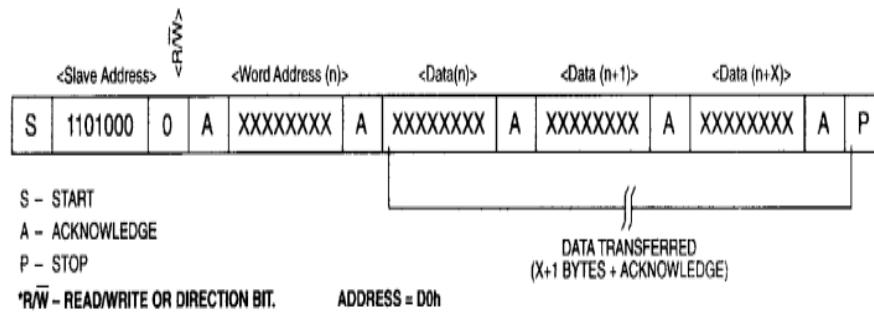
Berdasarkan bit R/W, maka ada 2 jenis transfer data yang mungkin :

**1. Transfer data dari master transmitter ke slave receiver :** byte pertama yang dikirim oleh master adalah slave address, dan selanjutnya adalah byte data. Slave akan kembali merespon bit setelah setiap byte diterima. Data ditransfer dengan bit (MSB) pertama yang paling signifikan.

**2. Transfer data dari slave transmitter ke master receiver :** Byte pertama (slave address) ditransmisikan menuju master. Slave lalu akan menjawab bit. Ini diikuti oleh transmisi slave yang berisi sejumlah byte data. Master selanjutnya akan merespon bit setelah semua byte diterima, selain byte terakhir.

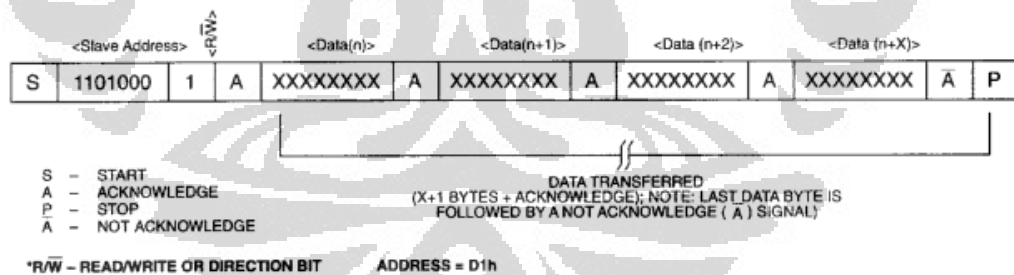
DS1307 beroperasi dengan mengikuti 2 mode :

1. **Mode Slave Receiver (DS1307 write mode) :** Data Serial dan clock diterima melalui SDA dan SCL. Setelah setiap byte diterima, lalu bit merespon dan ditransmisikan. Kondisi START dan STOP dikenal sebagai awal dan akhir dari transfer secara serial. Pengenalan Address dilakukan oleh hardware setelah menerima address slave (Lihat Gambar 8). Address byte adalah byte pertama yang diterima setelah kondisi START dihasilkan oleh master. Byte Address berisi 7 bit address DS1307 dengan bilangan biner 1101000, lalu selanjutnya diikuti oleh proses R / W yang menghasilkan penulisan/write 0. Setelah menerima dan mendecoding, address byte pada perangkat output akan dijawab pada line SDA. Setelah DS1307 merespon slave address dan menuliskan bit, master akan mentransmisikan register address ke DS1307. Hal ini akan mengatur register pointer pada DS1307. Master kemudian akan mulai mengirim setiap byte data dan akan merespon setiap byte yang diterima. Master akan menghasilkan kondisi STOP untuk menghentikan penulisan data.



Gambar 2.23 Data Write – Slave Receiver Mode

2. **Mode slave transmitter (DS1307 read mode)** : Byte pertama diterima dan diatur dalam mode slave receiver. Serial data dikirim ke SDA oleh DS1307, dimana serial clock sebagai input SCL. Kondisi START dan STOP dikenal sebagai awal dan akhir transfer serial (Lihat Gambar 9). Address byte adalah byte pertama yang diterima setelah kondisi START dieksekusi oleh master. Byte alamat berisi 7-bit register DS1307 dimana bilangan biner 1101000 diproses oleh R / W yang hasil pembacaannya adalah 1. Setelah menerima dan decoding, address byte perangkat input akan direspon pada line SDA. DS1307 kemudian akan mulai mengirimkan data dengan mengirimnya ke address register yang diarahkan oleh register pointer.



Gambar 2.24 Data Read – Slave Transmitter Mode

## 2.8 Komunikasi Serial

Komunikasi serial merupakan salah satu cara untuk mengkomunikasikan data dari suatu peralatan ke peralatan lain dengan cara menggunakan data secara serial. Misalnya mengkomunikasikan antara HP dengan Mikrokontroler, HP dengan PC, *printer* dengan PC dll. Pada PC, komunikasi serial RS232 dapat



dilakukan melalui port serial ( *COM port* ). Komunikasi data serial dapat dilakukan dengan mempresentasikan data dalam bentuk level “1” atau “0”.

*Interface* menggunakan port serial lebih sukar dilakukan jika dibandingkan dengan menggunakan port paralel. Pada umumnya setiap peralatan yang dihubungkan dengan menggunakan port serial harus mengubah kembali transmisi serial menjadi bentuk data paralel sebelum digunakan. Hal ini dapat dilakukan dengan menggunakan UART ( *Universal Asynchronous Receiver Transmitter* ). Namun ada beberapa kelebihan menggunakan komunikasi serial sehingga cukup sering digunakan, yaitu :

1. Kabel yang digunakan untuk komunikasi dapat lebih panjang. Port serial mengirimkan level “1” sebagai  $-3$  s/d  $-25$  volt dan level “0” sebagai  $+3$  s/d  $+25$  volt, sedangkan port paralel mengirimkan level “0” sebagai 0 volt dan level “1” sebagai 5 volt. Karena itu port serial memiliki swing maksimum 50 volts, sedangkan port paralel memiliki swing maksimum 5 volt. Hal ini menyebabkan kehilangan tegangan yang disebabkan oleh panjang kabel memiliki pengaruh yang lebih kecil pada komunikasi serial.
2. Kabel yang digunakan lebih sedikit. Komunikasi serial memiliki *null modem* sehingga hanya memerlukan 3 kabel, sedangkan pada komunikasi paralel memerlukan minimal 8 buah kabel data.
3. Pada peralatan yang menggunakan sinar infra merah sebagai media komunikasi penggunaan komunikasi data paralel tidaklah mungkin, karena sangat sulit membedakan sinar yang datang dari 8 bit.
4. Komunikasi serial mengurangi jumlah pin yang dibutuhkan pada mikrokontroler. Pada komunikasi serial hanya menggunakan dua buah pin, yaitu *Transmit Data* (TxD) dan *Receive Data* (RxD).

Peralatan yang menggunakan komunikasi serial dapat berupa DCE ( *Data Communication Equipment* ) atau DTE ( *Data Terminal Equipment* ). DCE seperti *modem, plotter, printer*, dll. Sedangkan DTE seperti PC dan terminal. Spesifikasi elektrik port serial dinyatakan pada EIA. Standar ini menetapkan persyaratan antara lain :

1. Logika 0 atau “Space” berupa tegangan  $+3$  dan  $+25$  volt

2. Logika 1 atau “Mark” berupa tegangan antara –3 dan –25 volt
3. Daerah antara –3 dan +3 volt tidak terdefinisi
4. Tegangan rangkaian terbuka ( open circuit voltage ) terhadap ground tidak melebihi 25 volt.
5. Arus pada hubungan singkat ( short circuit current ) tidak melebihi 500mA.

Konektor yang digunakan pada komunikasi serial data dengan menggunakan serial port yang dapat berupa konektor DB-25 atau DB-9. pada penulisan ini menggunakan DB-9. Berikut merupakan merupakan daftar fungsi pin pada DB-9.

**Tabel 2.4** Daftar Fungsi Pin DB-9

Pin DB-25	Pin DB-9	Tanda	Fungsi
8	1	DCD ( <i>Data Carrier Detect</i> )	Jika modem mendeteksi <i>carrier</i> dari modem lain maka DCD aktif
3	2	RXD ( <i>Receive Data</i> )	Serial data input
2	3	TXD ( <i>Transmit Data</i> )	Serial data output
20	4	DTR ( <i>Data Terminal Ready</i> )	Memberitahu bahwa UART siap
7	5	GND ( <i>Ground</i> )	Signal ground
6	6	DSR ( <i>Data Set Ready</i> )	Memberitahu UART bahwa modem siap
4	7	RTS ( <i>Request To Send</i> )	Menandakan UART siap bertukar data
5	8	CTS ( <i>Clear To Send</i> )	Menandakan modem siap bertukar data
22	9	RI ( <i>Ring Indicator</i> )	Aktif bila ada sinyal masuk



**Gambar 2.25** Urutan Letak Pin Female RS232 DB-9



**Gambar 2.26** Urutan Letak Pin Male RS232 DB9

Jika kecepatan pengiriman dan penerimaan data antara DTE dan DCE tidak sama maka akan terjadi luapan (*overflow*) pada *buffer* penerima data. Untuk menghindari hal ini maka diperlukan *flow control*. *Flow control* dapat dilakukan dengan dua cara, yaitu dengan *hardware* atau dengan *software*.

Secara *software* alur kendali biasanya dinyatakan sebagai Xon dan Xoff. Xon biasanya dinyatakan dengan karakter ASCII 17 sedangkan Xoff dinyatakan dengan karakter ASCII 19. Ketika *buffer* pada modem telah penuh, modem mengirimkan karakter Xoff sebagai tanda agar komputer menunda pengiriman data. Ketika *buffer* mempunyai ruang untuk menerima data, maka modem kemudian mengirimkan karakter Xon dan komputer dapat mengirim data kembali. Keuntungan *flow control* dengan *software* tidak memerlukan tambahan kabel karena pengiriman karakter Xon dan Xoff dilakukan melalui jalur TXD dan RXD. Namun dapat memperlambat proses komunikasi karena setiap karakter membutuhkan 10 bit.

*Flow control* secara *hardware* dikenal sebagai RTS/CTS *flow control*. Cara menggunakan dua buah kabel pada kabel serial menggantikan pengiriman tambahan karakter pada jalur data sehingga hal ini tidak memperlambat proses komunikasi data. Ketika komputer akan mengirim data maka komputer akan mengaktifkan jalur RTS. Jika modem mempunyai ruang untuk menerima data tersebut maka modem akan membalas dengan mengaktifkan jalur CTS dan kemudian komputer mulai mengirimkan data. Jika modem tidak mempunyai ruang untuk menerima data CTS tidak akan dikirimkan.

Komunikasi serial RS232 pada PC dilakukan melalui port serial COM *port*. Alamat COM *port* pada PC dapat dilihat pada tabel 2.9.

**Tabel 2.5 Standard Port Addresses DB-9**

Name	address	IRQ
COM 1	3F8	4
COM 2	2F8	3
COM 3	3E8	4
COM 4	2E8	3

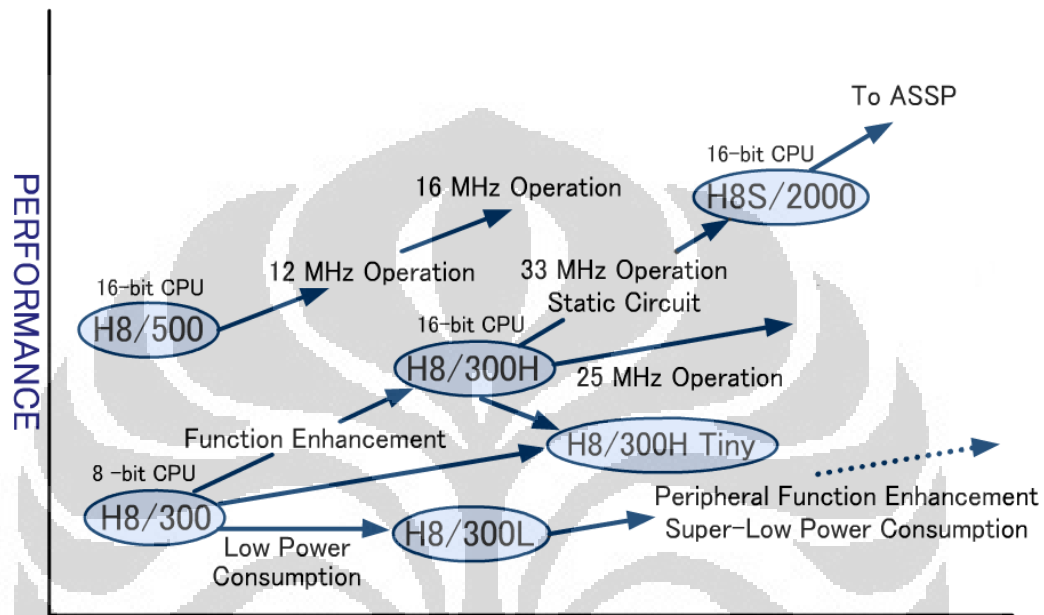
**Tabel 2.6 Daftar Register DB-9**

Base Address	DLAB	Read/Write	Abr.	Register Name
+0	=0	Write	-	Transmitter Holding Buffer
	=0	Read	-	Receiver Buffer
	=1	Read/Write	-	Divisor Latch Low Byte
+1	=0	Read/Write	IER	Interrupt Enable Register
	=1	Read/Write	-	Divisor Latch High Byte
+2	-	Read	IIR	Interrupt Identification Register
	-	Write	FCR	FIFO Control Register
+3	-	Read/Write	LCR	Line Control Register
+4	-	Read/Write	MCR	Modem Control Register
+5	-	Read	LSR	Line Status Register
+6	-	Read	MSR	Modem Status Register
+7	-	Read/Write	-	Scratch Register

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

## 2.9 Mikrokontroler H8/3069F

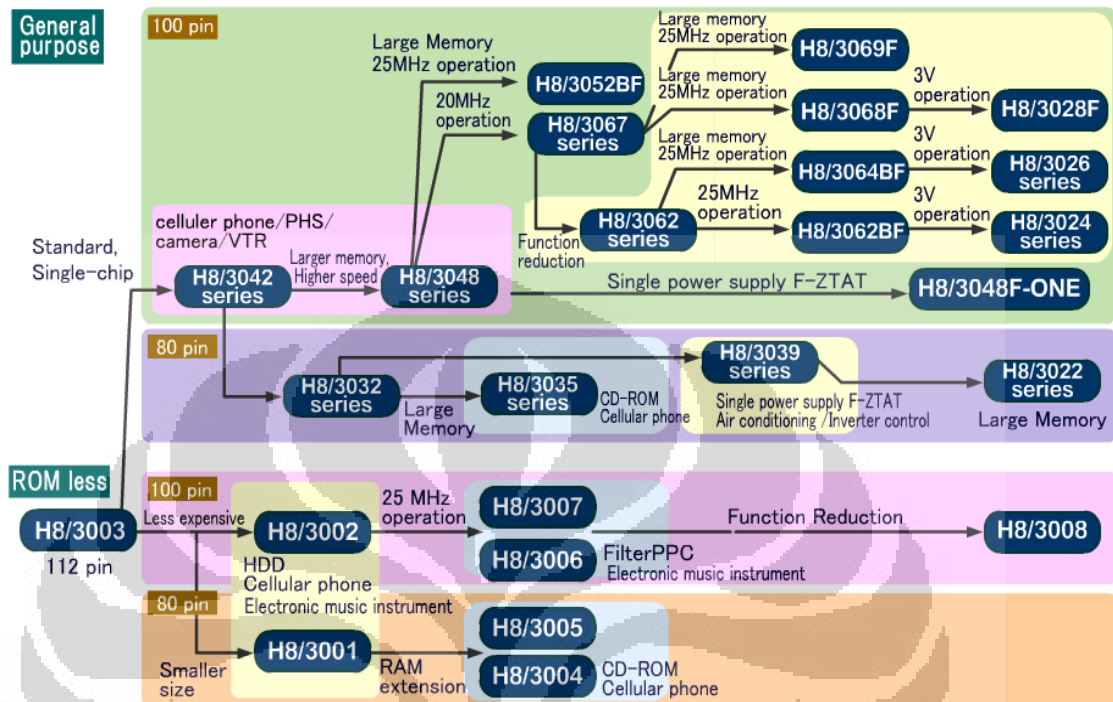
Seri *H8/300H* adalah single-chip microcomputer beformat tinggi yang mempunyai sebuah 16-bit CPU sebagai processornya. Gambar 2.11 akan menunjukkan daftar produk dari keluarga H8.



Gambar 2.27 Perkembangan Mikrokontroler H8.

Gambar 2.35 menunjukkan bagaimana seri H8 berkembang. Seri-seri H8 dibagi menjadi dua grup. Pertama adalah seri *H8/500*, yang menggunakan CPU 16-bit. Kedua adalah seri *H8/300*, yang menggunakan CPU 8-bit. Produk dari seri ini umumnya digunakan sebagai single-chip mikrokomputer. Seri *H8/300H* yang merupakan pengembangan dari *H8/300*, menggunakan CPU 16-bit dan memungkinkan koneksi memori hingga sebesar 16M byte. Produk dari seri ini tidak hanya digunakan sebagai single-chip mikrokomputer, tetapi juga dapat digunakan sebagai multi-chip mikrokomputer yang memungkinkan pemberian memori eksternal. Seri *H8S/2000* dikembangkan dengan menambah instruksi dari seri *H8/300H*. *H8S/2000* lebih stabil dan memungkinkan operasi berkecepatan tinggi sampai 33 MHz. Pada daftar H8 juga terdapat seri *H8/300L* yang merupakan versi *low power consumption* dari seri *H8/300*. Pada daftar H8 juga terdapat seri *H8/300H tiny* yang merupakan versi *compact* dari seri *H8/300H*.

Mikrokontroler seri *H8/300H* umumnya digunakan sebagai sarana edukasi di perguruan tinggi.

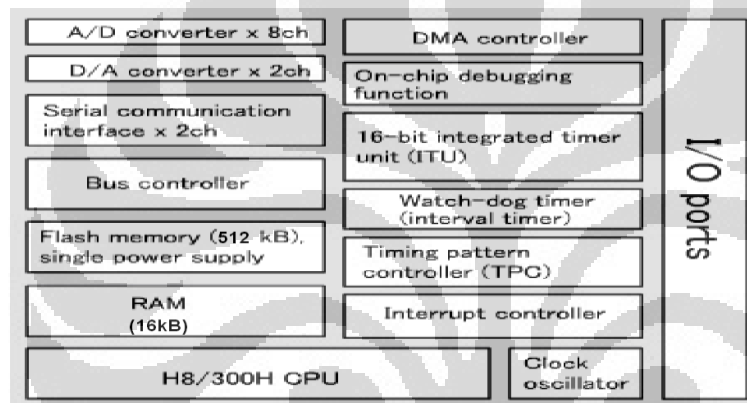


Gambar 2.28 Daftar perkembangan mikrokontroler yang menggunakan mikroprocessor *H8/300H*

Gambar 2.28 menunjukkan perkembangan dari seri *H8.300H* dan bidang aplikasi serta fitur-fiturnya. Produk pertama dari seri *H8/300H* adalah *H8/3003*. *H8/3003* tidak bisa digunakan sebagai single-chip mikrokomputer, karena tidak punya ROM didalamnya. *H8/3042* mempunyai ROM dan bisa digunakan sebagai single-chip mikrokomputer. *H8/3069F* adalah versi pengembangan dari pendahulunya. *H8/3069F* telah memiliki memori yang diperbesar untuk peningkatan kecepatan. *FZTAT* mengindikasikan bahwa flash memori telah terintegrasi didalamnya. Tipe flash memori ini adalah EEPROM yang berarti dapat dihapus secara elektrik dan diprogram kembali. Register-register seperti pada telepon selular atau alat-alat lainnya telah tertulis di flash memori. Karena flash memori adalah ROM maka data dapat disimpan tidak akan hilang walaupun powernya telah mati. Karena

sifatnya yang dapat diprogram, maka ROM dapat dimasukkan data. Pada pengembangannya terdapat berbagai macam produk lainnya yang fungsi, kapasitas memori, harga dan kecepatan operasinya beragam.

Mikrokontroler *H8/3069F* adalah salah satu mikrokontroler yang menggunakan processor *H8/300H*. *H8/3069F* mempunyai sebuah internal writable flash memory yang menggunakan single power supply (5V). Perbedaannya dengan keluarga mikrokontroler H8 yang memakai microprocessor *H8/300H* lainnya adalah kapasitas ROM dan RAM-nya dan fitur lainnya. Gambar 3.11 menunjukkan fitur-fitur pada mikrokontroler *H8/3069F*.



Gambar 2.29 Fitur-fitur dari mikrokontroler *H8/3069F*

#### Fitur-fitur dari CPU

- 16-bit CPU yang berperan sebagai general-purpose register

Dilengkapi dengan 16-bit x 16 general-purpose register. Dan tersedia juga dalam 8-bit x 8 + 16-bit x 16 atau 32-bit x 8.

- CPU berkecepatan tinggi

Frekuensi operasi maksimum dari *H8/3069F* adalah 25MHz. Fungsi penambahan/pengurangan dapat dieksekusi dalam 80ns dan perkalian/pembagian dieksekusi dalam 560ns.

Operasi CPU berbasis pada sinyal clock, semakin tinggi frekuensi sinyal clock-nya, semakin cepat operasinya. Waktu dari pulsa sinyal clock 25MHz adalah 0.04 microsecond (40ns), yang disebut “1 state”. Penambahan/pengurangan diselesaikan dalam “2 state” sedangkan pengalihan/pembagian diselesaikan dalam “14 state”.

- Dilengkapi dengan ruang address maksimum sebesar 16M byte

### **Berfungsi sebagai single-chip dan multi-chip mikrokomputer**

Mikrokontroler ini dapat difungsikan sebagai single-chip mikrokomputer karena internal ROM, RAM dan fungsi I/O pada CPU. Mikrokontroler ini juga dapat difungsikan sebagai multi-chip mikrokomputer saat terjadi penambahan memori.

#### **Internal ROM**

Mikrokontroler ini mempunyai flash memory 512K byte yang dapat di write dengan sebuah power supply (5V).

#### **Internal RAM**

Mikrokontroler ini mempunyai 16K byte internal RAM.

#### **I/O port: 11 I/O port**

I/O port dapat digunakan sebagai input status on/off atau sinyal dari berbagai sensor. Saat I/O port digunakan sebagai output, mikrokontroler dapat diatur untuk mengontrol kedipan lampu atau mengontrol switch on/off dari motor dan heater.

#### **Internal SCI (Serial Communication Interface) x 3 channel**

Ketiga channel SCI mempunyai fungsi yang sama. Mode dari SCI ini adalah sinkron dan asinkron. SCI mikrokontroler ini juga mempunyai komunikasi



multiprocessor dengan dua atau lebih processor. SCI juga dapat dihubungkan dengan smart card interface dengan mengubah setting pada register CPU.

### **Internal 16-bit timer x 3 channel dan 8-bit timer x 4 channel**

Channel 0 dan 1 pada 16-bit timer mempunyai fungsi yang sama, sedangkan channel 2 mempunyai register sendiri pada CPU. 8-bit timer dibagi menjadi dua grup dengan masing-masing dua channel. Grup 0 terdiri dari channel 0 dan 1 dan grup 1 terdiri dari channel 2 dan 3.

### **Internal TPC (Timing Pattern Controller)**

*H8/3069F* mempunyai TPC yang menyediakan output pulsa dengan berbasis 16-bit timer. Pulsa output dari TPC dibagi menjadi grup 4-bit (grup 3 sampai grup 0) yang dapat beroperasi secara serempak dan independent.

### **Internal watch-dog timer (WDT)**

WDT dapat dioperasikan untuk mengawasi jalannya program, atau hanya sebagai interval timer. Ketika WDT digunakan, WDT akan membangkitkan sinyal reset pada chip *H8/3069F* bila sistem crash.

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

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

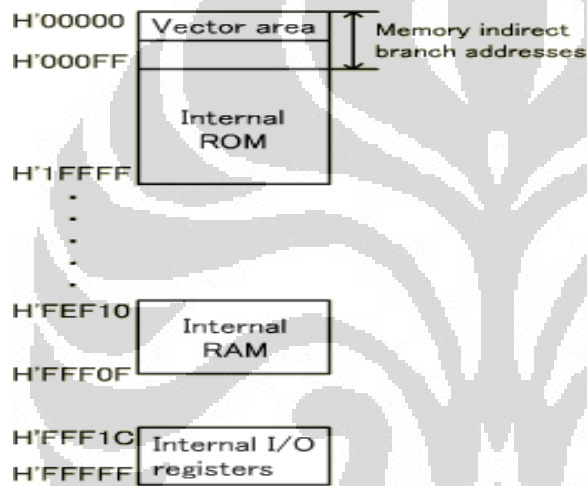
### **Internal 8-bit D/A converter × 2 channel**

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.

### Internal DMA controller × 4 channel

Digunakan untuk transfer data berkecepatan tinggi. DMAC memungkinkan transfer data lebih cepat dari penggunaan CPU. Umumnya digunakan dengan sebuah timer dan fungsi komunikasi lainnya.

*H8/3069F* dapat digunakan sebagai single-chip mikrokomputer. Pada kondisi ini, hanya internal memori yang dapat digunakan. Gambar 3.12 menunjukkan memori map pada mode single-chip. Pada mode ini alamat memori diekspresikan dengan notasi 5 digit heksadesimal.



Gambar 2.30 Memori map mode single-chip

## 2.10 Python

Python merupakan bahasa pemrograman yang bersifat *object-oriented*. Bahasa Python dibuat pada tahun 1980an yang namanya diambil dari salah satu tayangan televisi di Inggris yaitu *Monty Python's Flying Circus*. Meskipun Python tidak terlalu terkenal dibandingkan dengan bahasa pemrograman lainnya, Python dapat diandalkan untuk membuat software aplikasi di bidang sains dan teknik dengan efisien dan elegan.

Python memiliki kelebihan lain yang sangat penting dibanding bahasa pemrograman yang terdahulu:

- a. Python merupakan *open-source software*, yang artinya ia dapat diperoleh secara gratis. Bahkan Python sudah otomatis terinstall di Linux.
- b. Python tersedia pada semua *operating systems* (OS) terkenal seperti Linux, Unix, Windows, dan MacOS. Suatu *script* Python yang ditulis pada OS tertentu, dapat dijalankan di OS lain tanpa ada modifikasi sedikitpun.
- c. Python lebih mudah dipelajari sekaligus lebih mudah "dibaca" dibandingkan dengan bahasa pemrograman lainnya.
- d. Python dan program ekstensinya mudah diinstall.

Python berdiri di atas landasan pondasi Java and C++. Hal-hal seperti *classes*, *methods*, *inheritance*, yang kerap kali diimplementasikan pada bahasa yang bersifat *object-oriented*, juga dapat diimplementasikan di Python. Python merupakan salah satu bahasa pemrograman tingkat tinggi. Sintaks inti Python dan strukturnya merupakan kode yang mudah dibaca dengan *library* yang luas dan dapat disesuaikan penggunaannya (Norton, 2005)

## 2.11 Database MySQL

Pada penelitian ini digunakan basis data MySQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data *Structured Query Language* (SQL) atau *Database Management System* (DBMS) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL [21].

Sebagai *database* server, MySQL dapat dikatakan lebih unggul dibandingkan *database* server lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain [21] :

- *Portability*, yaitu MySQL dapat berjalan stabil pada berbagai sistem operasi

seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

- *Multiuser*, yaitu MySQL dapat digunakan oleh beberapa pengguna atau *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- *Performance tuning*, yaitu MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- *Security*, yaitu MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.
- *Scalability dan limits*, yaitu MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

Bahasa pemrograman, terdapat beberapa API tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses *database* MySQL antara lain bahasa pemrograman C, C++, C#, bahasa pemrograman Eiffel, bahasa pemrograman Smalltalk, bahasa pemrograman Java, bahasa pemrograman Lisp, Perl, PHP, bahasa pemrograman Python, Ruby, REALbasic dan Tcl. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C.

## BAB 3

### PERANCANGAN ALAT

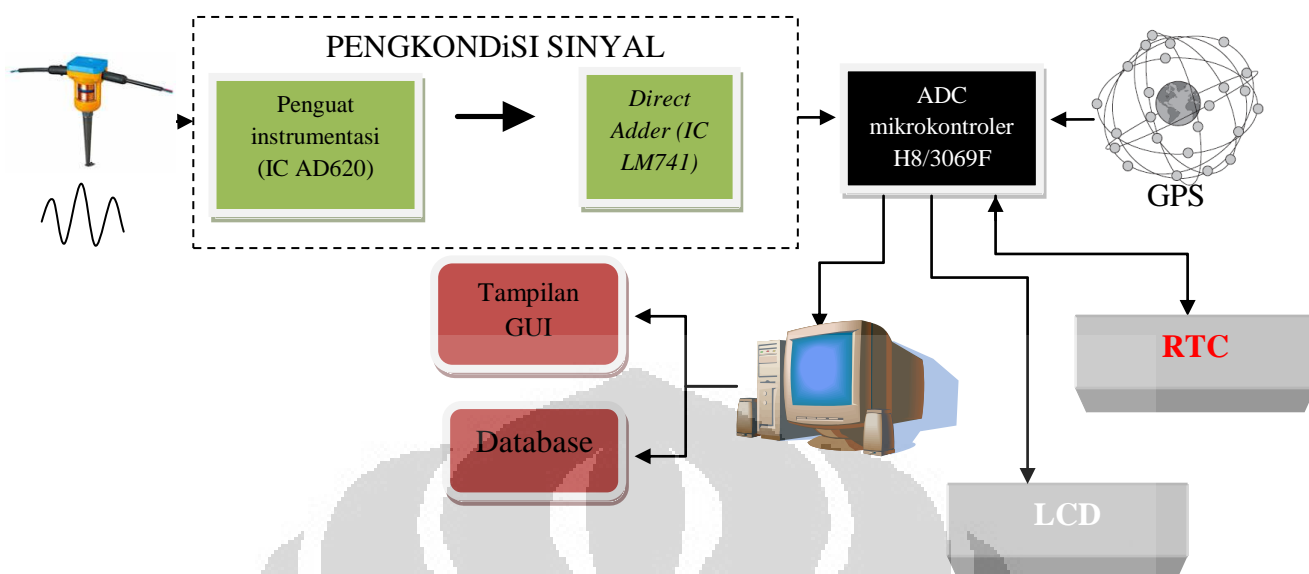
#### 3.1 PERANGKAT KERAS

Pada kegiatan eksplorasi hidrokarbon maupun sumber daya alam lain yang terkandung dibawah permukaan bumi, gelombang seismik yang digunakan adalah gelombang seismik buatan. Sumber gelombang seismik buatan tersebut dapat berasal dari ledakan dinamit, *vibroseis*, benda jatuh, ataupun sumber getaran lainnya seperti pukulan palu ke tanah seperti yang diterapkan pada penelitian ini.

Metode yang digunakan pada eksplorasi seismik ini pada dasarnya adalah memanfaatkan gelombang pantul (refleksi) ataupun gelombang bias (refraksi) yang berasal dari sumber getaran dan ditangkap oleh *receiver* (geophone) di permukaan bumi (Purwanto, 2007).

Sinyal atau gelombang seismik yang ditangkap oleh geophone adalah sinyal yang relatif lemah dibandingkan dengan sumber sinyal yang dikirimkan. Oleh karena itu, diperlukan suatu pengkondisi sinyal yang dapat memperkuat sinyal tersebut agar dapat dibaca dan diolah oleh blok pengkondisi sinyal selanjutnya.

Pada perancangan sistem logger dan akuisisi data gelombang seismik ini digunakan penguat instrumentasi sebagai blok sistem yang akan memperkuat sinyal seismik yang diterima geophone. Penguat instrumentasi merupakan penguat yang memiliki penguatan besar yang dapat diatur secara akurat, impedansi masukan dan CMRR (*Common Mode Rejection Ratio*) tinggi, tegangan offset rendah, dan lain sebagainya. Sinyal ini kemudian akan dikonversi oleh ADC internal mikrokontroler H8/3069F yang memiliki rentang batas pembacaan tegangan 0-5V dc, sehingga diperlukanlah rangkaian *direct adder* guna mengkondisikan sinyal yang ditangkap geophone. Hasil konversi ini barulah disimpan pada memori dan ditampilkan pada GUI Phyton.



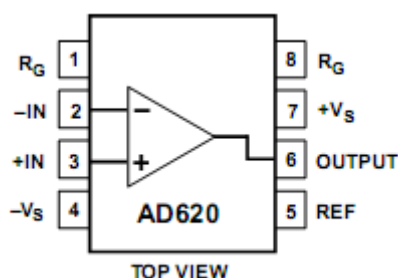
Gambar 3.1 Blok diagram seismic data logger tersinkronisasi waktu GPS

### 3.1.1 Rangkaian Penguat Instrumentasi

Rangkaian penguat instrumentasi pada penelitian ini dirancang dengan menggunakan satu buah *operational amplifier* yang terdapat pada IC (*Integrated Circuit*) AD620 buatan Analog Devices.

Beberapa karakteristik op-amp pada IC AD620, yaitu :

- Setting penguatan dengan 1 resistor eksternal ( Gain 1 sampai 1000)
- Interval tegangan power supply 2,3 V – 18V
- Adanya proteksi hubungan singkat,
- Impedansi input tinggi,
- Kompensasi frekuensi internal,
- Daya dan noise yang rendah



Gambar 3.2 Konfigurasi pin IC AD620

Penguat instrumentasi ini terdiri atas satu buah op-amp. Besarnya penguatan yang digunakan pada perancangan sistem data logger ini adalah sebesar 1000, sehingga sangat praktis apabila menggunakan potensiometer pada RG ( hambatan penguat).

### 3.1.2 Rangkaian *Direct Adder*

Dalam penelitian ini peran rangkaian *Direct Adder* adalah rangkaian yang bertujuan untuk menambahkan tegangan dc ke sinyal masukan. Penggunaan rangkaian *Direct Adder* pada penelitian ini sangat penting sebagai pengkondisi sinyal sebelum melewati ADC (*Analog to Digital Converter*) internal mikrokontroler H8/3069F. Hal ini dikarenakan ADC internal mikrokontroler H8/3069F adalah ADC 10 bit dengan rentang masukan 0-5V dc. IC yang digunakan untuk direct adder ini adalah IC yang banyak terdapat di pasaran, yaitu IC LM741.

### 3.1.3 Pengkonversi Analog ke Digital Internal Mikrokontroler H8/3069F

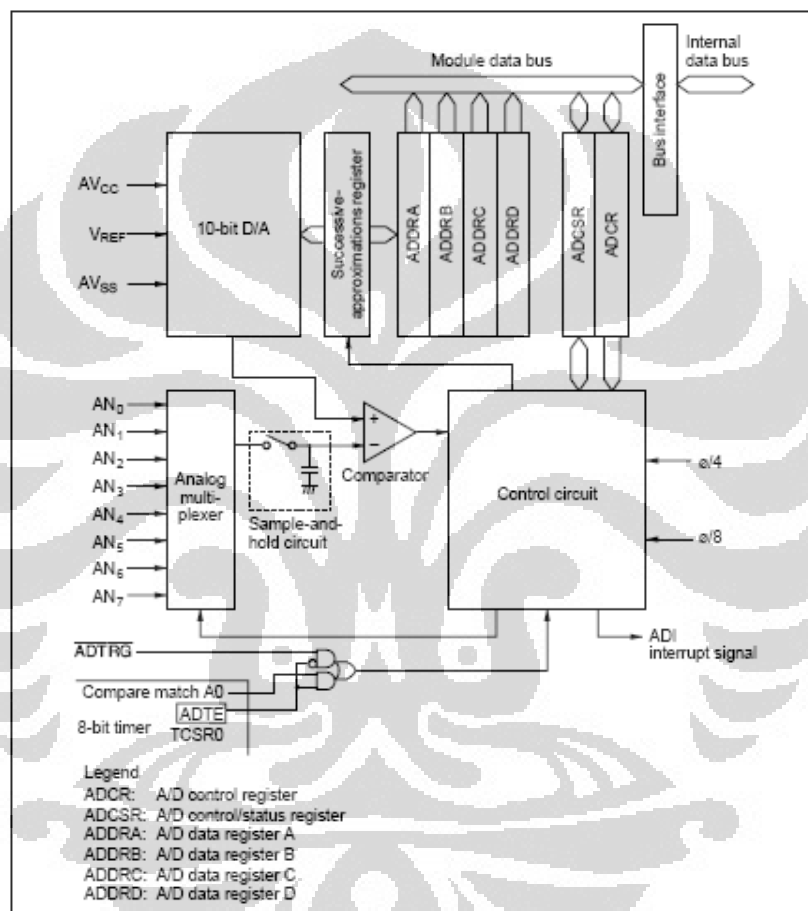
Sebelum diproses oleh sebuah *processor* (mikrokontroler), pada umumnya masukan sinyal analog yang telah melewati suatu pengkondisi sinyal dikonversi terlebih dahulu menjadi data digital. Konversi ini dilakukan oleh ADC (*Analog to Digital Converter*). Analog to Digital Converter adalah suatu rangkaian 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 (8) buah kanal input konversi sehingga menjadikan ADC ini sangat tepat diaplikasikan untuk sistem akuisisi data dengan banyak input.

Beberapa fitur ADC internal mikrokontroler H8/3069F adalah :

- Resolusi 10 bit,
- Memiliki 8 kanal input,
- Waktu konversi yang cepat (maksimum 3.5  $\mu$ s per kanal),
- Memiliki 2 mode konversi, yaitu : Mode single (mengkonversi satu kanal) dan mode scan (mengkonversi banyak kanal sekaligus)

- Batasan konversi tegangan input analog dapat ditentukan berdasarkan tegangan referensi pada pin  $V_{REF}$ ,
- Memiliki empat data register 16 bit,
- Terdapat fungsi *sample and hold*,
- Memiliki 3 sumber pemicu dimulainya konversi, yaitu dengan software, pemicu eksternal, dan timer 8 bit,
- *A/D end interrupt* (ADI).



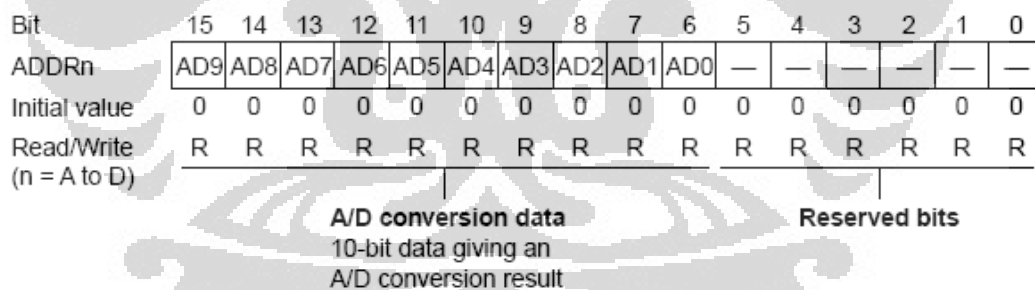
Gambar 3.3 Blok diagram ADC internal mikrokontroler H8/3069F

ADC internal mikrokontroler H8/3069F memiliki empat buah data register yaitu ADDRA sampai ADDR D 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.

Table 3.1 Konfigurasi register ADC internal mikrokontroler H8/3069F



Address	Name	Simbol	R/W	Initial value
H'FFFE0	A/D data register A H	ADDRAH	R	H'00
H'FFFE1	A/D data register A L	ADDRAL	R	H'00
H'FFFE2	A/D data register B H	ADDRBH	R	H'00
H'FFFE3	A/D data register B L	ADDRBL	R	H'00
H'FFFE4	A/D data register C H	ADDRCH	R	H'00
H'FFFE5	A/D data register C L	ADDRCL	R	H'00
H'FFFE6	A/D data register D H	ADDRDH	R	H'00
H'FFFE7	A/D data register D L	ADDRDL	R	H'00
H'FFFE8	A/D control/status register	ADCSR	R/(W)*	H'00
H'FFFE9	A/D control register	ADCR	R/W	H'7E



Gambar 3.4 Read only register 16 bit

ADC H8/3069F menyediakan delapan pin kanal input yang dibagi menjadi dua grup, yaitu :

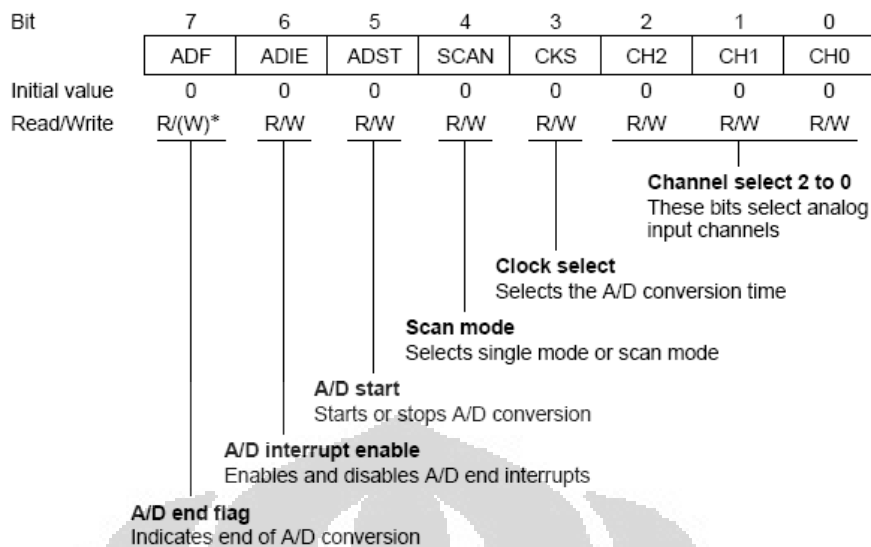
- Grup 0 : terdiri atas AN0-AN3, dan
- Grup 1 : terdiri atas AN4-AN7.

Tabel 3.2 Konfigurasi pin ADC internal mikrokontoler H8/3069F

Pin	Simbol	I/O	Fungsi

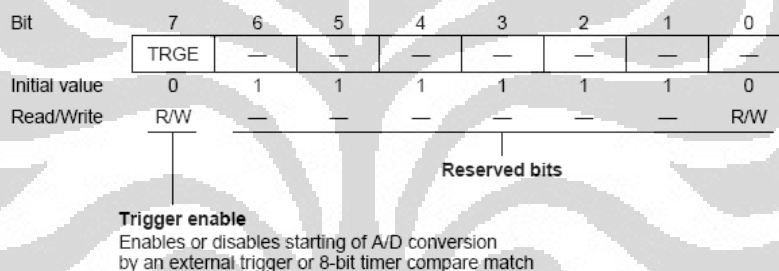
Power suplai	AVcc	Input	Power suplai analog
Analog ground	Avss	Input	Analog ground
Tegangan referensi	Vref	Input	Tegangan referensi
Input analog 0	AN0	Input	Input analog grup 0
Input analog 1	AN1	Input	
Input analog 2	AN2	Input	
Input analog 3	AN3	Input	
Input analog 4	AN4	Input	Input analog grup 1
Input 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.



Note: \* Only 0 can be written, to clear the flag.

Gambar 3.6 A/D control/status register (ADCSR)



Gambar 3.5 A/D control register (ADCR)

### 3.1.4 Modul GPS GARMIN

Modul GPS yang digunakan dalam perancangan seismik data logger ini adalah GARMIN 18x PC LVC. Interface pada tipe GPS ini dilakukan dengan menggunakan serial port dari komputer via konektor DB9.



Gambar 3.6 GPS GARMIN

Beberapa fitur GPS Garmin 18x PC 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)

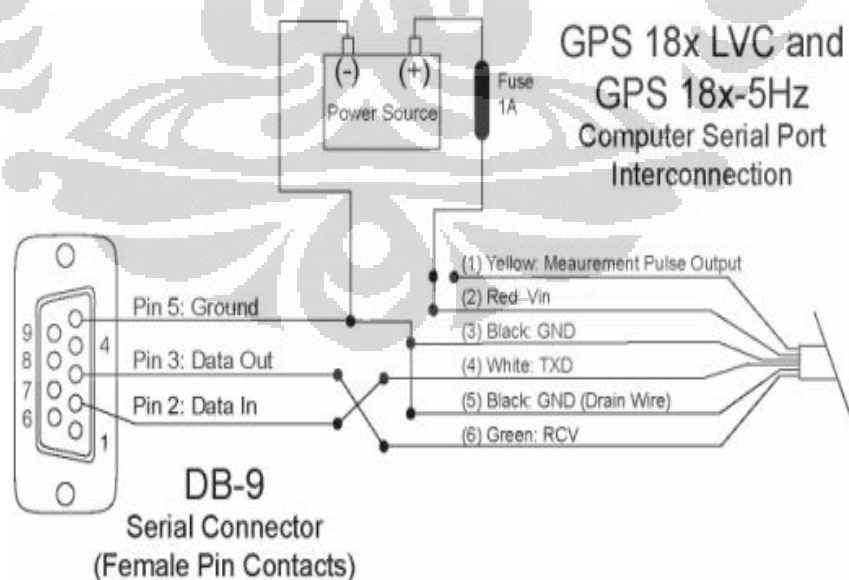
Posisi : < 3 meter, 95% typical

Kecepatan : 0,1 knot RMS ready state

- Komunikasi protokol dengan NMEA (*National Marine Electronics Association*)

Tabel 3.3 Deskripsi Pin Garmin

GPS 18x Pin #	Color	Signal Name	Wire Gauge
1	Yellow	Measurement Pulse Output	28
2	Red	Vin	26
3	Black	Ground	28
4	White	Transmit Data	28
5	Black	Ground	26
6	Green	Receive Data	28



Gambar 3.7 Interkoneksi Serial Port Komputer

Standar protokol interface yang digunakan untuk GPS ini adalah NMEA0183. Untuk mendapatkan waktu real time GPS yang nantinya akan di olah oleh mikro untuk ditampilkan oleh PC dan LCD diperlukan adanya sentence atau pernyataan dari GPS tersebut, yaitu \$GPRMC. \$GPRMC ini memuat informasi tidak hanya tanggal dan waktu saja, tetapi juga posisi, latitude, dan lain lain.

Tabel 3.4 Data \$GPRMC

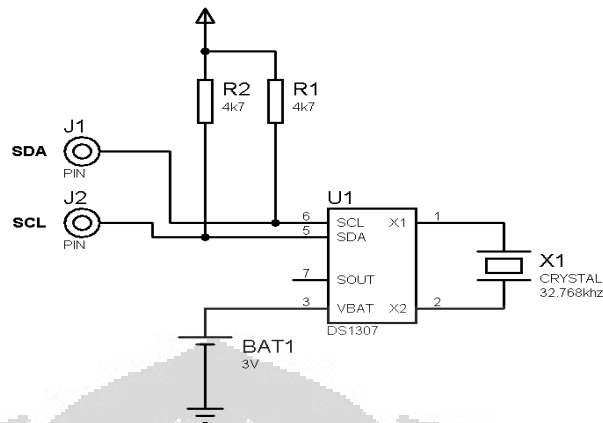
\$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12>\*hh<CR><LF>

<1>	UTC time of position fix, hhmmss format for GPS 18x PC/LVC; hhmmss.s format for GPS 18x-5Hz
<2>	Status, A = Valid position, V = NAV receiver warning
<3>	Latitude, ddmm.mmmm format for GPS 18x PC/LVC; ddmm.mmmmm format for GPS 18x-5Hz (leading zeros must be transmitted)
<4>	Latitude hemisphere, N or S
<5>	Longitude, dddmm.mmmm format for GPS 18x PC/LVC; dddmm.mmmmm format for GPS 18x-5Hz (leading zeros must be transmitted)
<6>	Longitude hemisphere, E or W
<7>	Speed over ground, GPS 18x PC and LVC: 000.0 to 999.9 knots, GPS 18x-5Hz: 000.00 to 999.99 knots (leading zeros will be transmitted)
<8>	Course over ground, 000.0 to 359.9 degrees, true (leading zeros will be transmitted)
<9>	UTC date of position fix, ddmmyy format
<10>	Magnetic variation, 000.0 to 180.0 degrees (leading zeros will be transmitted)
<11>	Magnetic variation direction, E or W (westerly variation adds to true course)
<12>	Mode indicator (only output if NMEA 0183 version 2.30 active), A = Autonomous, D = Differential, E = Estimated, N = Data not valid

### 3.1.5 RTC (Real Time Clock)

RTC merupakan bagian dari sistem yang dapat menghitung dan mengatur waktu baik dalam detik, menit, jam, hari, tanggal, bulan dan tahun. Untuk dapat mengaktifkan clock dari DS1307 digunakan eksternal kristal sebagai pembangkit dengan filter 2 buah resistor R3 dan R4 setelah dilewati oleh tegangan Vcc 5V.

Adapun skematik rangkaian adalah seperti gambar berikut :



Gambar 3.8 Skematik rangkaian RTC

Dengan pengaturan bit pada alamat register-register pada DS1307 tersebut maka register control yang berperan menyeleksi bit digunakan untuk mengambil clock dan mengaktifkan RTC dalam IC DS1307 tersebut.

### 3.1.6 LCD 16x2 LMB162A

Untuk melihat display waktu hasil sinkronisasi pada hardware digunakan LCD 16x2 LMB162A. LCD jenis ini dapat menampilkan jumlah karakter sebanyak 16 karakter dalam 2 baris dengan ukuran karakter 2,95 x 5.55. Untuk mengatur intensitas cahaya yang ditampilkan, pada perancangan display ini digunakan trimpot. LCD ini nantinya akan dihubungkan dengan port register mikrokontriller H8/3069F



Gambar 3.9 LCD LMB162A

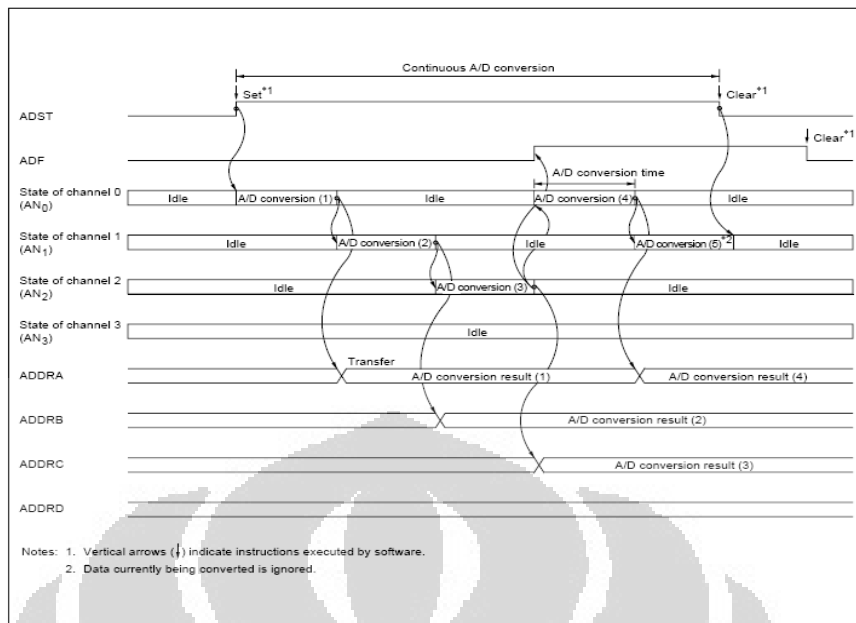
## 3.2 PERANGKAT LUNAK

Perangkat lunak terdiri dari pemrograman ADC mikrokontroler H8/3069F dengan bahasa C dan GUI (*Graphical User Interface*) berbasis bahasa pemrograman Python serta pemrograman database MySQL. Pemrograman ADC internal mikrokontroler H8/3069F yang dirancang adalah mode scan mode yang dapat mengkonversi delapan kanal input sekaligus, sesuai dengan jumlah sensor yang digunakan. Setelah dikonversi, data tersebut akan dikirim ke PC melalui koneksi serial dan ditampilkan pada GUI serta disimpan pada database MySQL.

### 3.2.1 Pemrograman ADC Internal Mikrokontroler H8/3069F

ADC H8/3069F adalah ADC yang menggunakan prinsip *successive approximations* dengan resolusi 10 bit. ADC ini memiliki dua mode operasi, yaitu *single mode* dan *scan mode*. *Scan mode* sangat berguna untuk memonitoring input analog dalam satu grup atau banyak kanal seperti pada penelitian ini.

Pada logaritma pemrograman ADC internal mikrokontroler H8/3069F mode scan, pertama-tama kita harus memberikan logika 1 pada bit ADCSR SCAN, kemudian pilih grup kanal input dan memberikan logika 1 pada ADST baik secara software maupun pemicu eksternal. Konversi analog ke digital akan dimulai dari kanal input pertama pada grup 0 (AN0 ketika kanal CH2 = 0, AN4 ketika CH2 = 1). Konversi ini akan dilanjutkan pada kanal selanjutnya yang dipilih hingga bit ADST bernilai 0. Data hasil konversi ini kemudian akan disimpan pada data register sesuai dengan kanalnya.



Gambar 3.10 Alur operasi ADC mode scan

Pemrograman ADC ini menggunakan bahasa pemrograman C yang kemudian di-*compile* menjadi file.mot dengan program WinAVR. Komunikasi dan pemrograman mikrokontroler H8/3069F dilakukan secara serial. Untuk memasukkan sebuah program ke dalam mikrokontroler H8/3069F, pertama-tama kita harus mengkoneksikan serial mikrokontroler dengan PC. Setelah itu, kita dapat memprogramnya dengan mencari file.mot yang sudah kita buat sebelumnya pada *command prompt*. Kemudian memprogramnya dengan menulis `h8write.exe -3069f -f20 (file.mot) com9`.

Sebelum melakukan *writing* ke modul mikrokontroler H8/3069F, aturlah posisi *DIP switch* ke mode-3 seperti pada Tabel 3.1 sebagai berikut:

Tabel 3.5 Posisi DIP Switch Mode-3

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



```

C:\ Command Prompt
F:\H8>cd reneasas
F:\H8\RENEASAS>cd akuisisi
F:\H8\RENEASAS\akuisisi>h8write.exe -3069 -f20 main.mot com1
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.
F:\H8\RENEASAS\akuisisi>h8write.exe -3069 -f20 test.mot com1
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.
F:\H8\RENEASAS\akuisisi>

```

Gambar 3.11 Langkah-langkah *writing* program ADC Internal

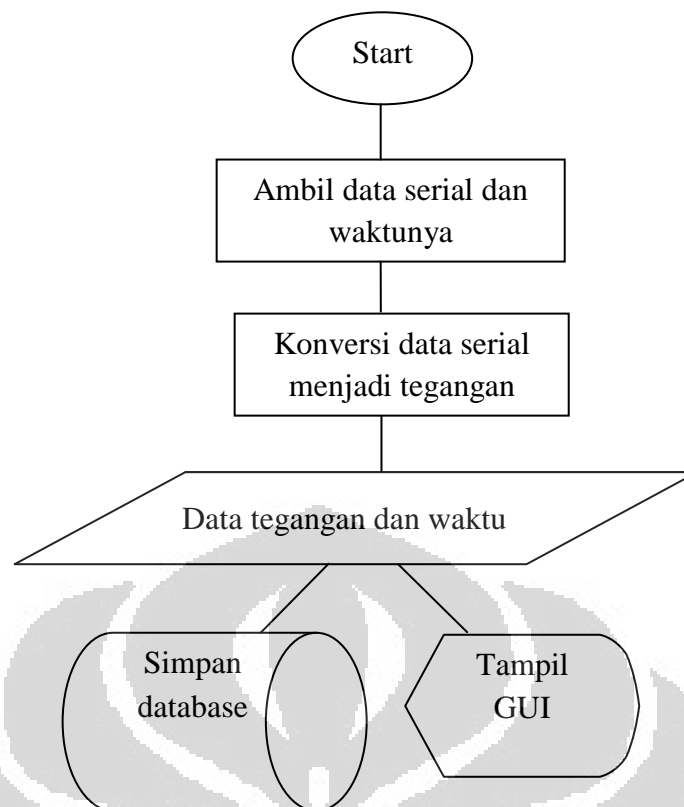
Setelah melakukan *writing* ubahlah posisi *DIP switch* ke mode-7 untuk melakukan *reading* seperti yang ditunjukkan pada Tabel 3.2

Tabel 3.6 Posisi DIP Switch Mode-7

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

### 3.2.2 Pemograman Tampilan GUI (*Graphical User Interface*) Phyton

Perangkat lunak pada GUI sistem akuisisi data gelombang seismik ini menggunakan bahasa pemograman Python yang dikategorikan sebagai bahasa tingkat tinggi. *Library* Python yang digunakan untuk membuat tampilan GUI pada penelitian ini adalah wxPython sedangkan *Library* yang digunakan untuk membaca data dari *database* pada penelitian ini adalah MySQL –Python yang terformat dalam file Microsoft excel. Diagram alur dari pembuatan GUI ditunjukkan pada gambar dibawah.



Gambar 3.12 Flowchart tampilan GUI Python

### 3.2.3 Penyimpanan Data

Database yang digunakan untuk menampung data dari satu buah sensor *geophone* adalah MySQL - Python. Data tersebut berupa besarnya nilai amplitudo gelombang seismik dan waktu pengambilannya. Hasil yang sudah ditangkap kemudian ditampung pada database dan dapat langsung ditampilkan pada GUI secara *real time*.

*Database* yang dibuat nantinya dapat menyimpan data tanggal dan waktu pengambilan data serta data amplitudo satu kanal input. Data ini kemudian dapat di simpan di dalam memori komputer, baik komputer yang langsung dihubungkan dengan sistem akuisisi (komputer lokal) maupun komputer server yang memiliki jaringan dengan komputer lokal.

## BAB 4

### ANALISIS HASIL PENELITIAN

#### 4.1 PERANGKAT KERAS

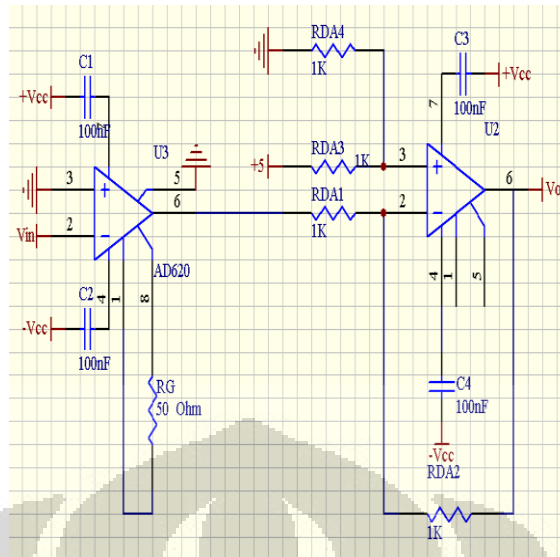
Pada sistem logger data gelombang seismik ini terdiri dari blok penguat, rangkaian *direct adder*, ADC internal mikrokontroler H8, RTC, GPS dan LCD. Setelah pengerjaan seluruh sistem, maka perlu dilakukan pengujian alat serta analisa terhadap alat dan data, apakah sistem sudah bekerja dengan baik atau tidak. Dalam proses perancangannya pertama-tama dilakukan pengujian terhadap masing-masing blok untuk mendapatkan karakteristik dan menguji fungsi blok tersebut. Pengujian tersebut meliputi

1. Uji rangkaian penguat instrumentasi.
2. Uji linearitas rangkaian ADC (Analog to Digital Converter) internal mikrokontroler H8/3069F.
3. Uji sistem waktu

##### 4.1.1 Uji Rangkaian Penguat Instrumentasi

Pada perancangan sistem logger data seismik ini terdiri dari rangkaian penguat yang berfungsi untuk menguatkan sinyal seismik. Rangkaian penguat yang digunakan adalah penguat instrumentasi dengan penguatan 1000x ( $A = 1000$ ).

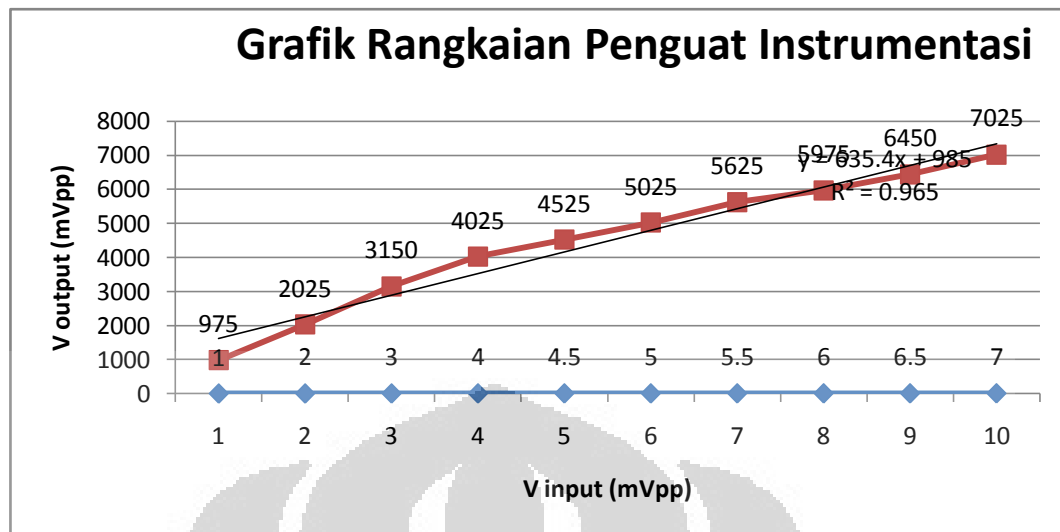
Pengambilan data dilakukan dengan perancangan dan pengaturan terhadap setiap komponen rangkaian seperti pada perancangan alat. Dengan memvariasikan besarnya nilai input yang diberikan dengan rentang 1-7 mVpp lalu melihat keluarannya pada osiloskop.



Gambar 4.1 Rangkaian Penguat Instrumentasi dan Direct Adder

Table 4.1 Data penguatan rangkaian penguat instrumentasi

No.	Vpp_input(mV)	Vpp_out(mV)	Penguatan,A
1	1	975	975
2	2	2025	1012.5
3	3	3150	1050
4	4	4025	1006.25
5	4.5	4525	1005.55
6	5	5025	1005
7	5.5	5625	1022.72
8	6	5975	995.84
9	6.5	6450	992.31
10	7	7025	1003.57
A rata-rata			1006.87



Gambar 4.1. Grafik Input Vs Output Rangkaian Penguat Instrumentasi

Data pengujian menunjukkan bahwa penguatan yang didapatkan adalah sebesar 1006.87. Penguatan ini sedikit tidak sesuai dengan penguatan yang diinginkan, yaitu sebesar 1000x. Hal ini bisa disebabkan oleh nilai toleransi resistor yang digunakan serta penggunaan variabel resistor yang digunakan. Pada pengujian, peneliti menggunakan resistor dengan toleransi 5% dan potensiometer dengan hambatan maksimum  $56\Omega$  yang sensitivitasnya kurang baik.



Gambar 4.2 Input dan output rangkaian penguatan instrumentasi

#### 4.1.2 Uji Linearitas Rangkaian ADC (Analog to Digital Converter) Internal Mikrokontroler H8/3069F

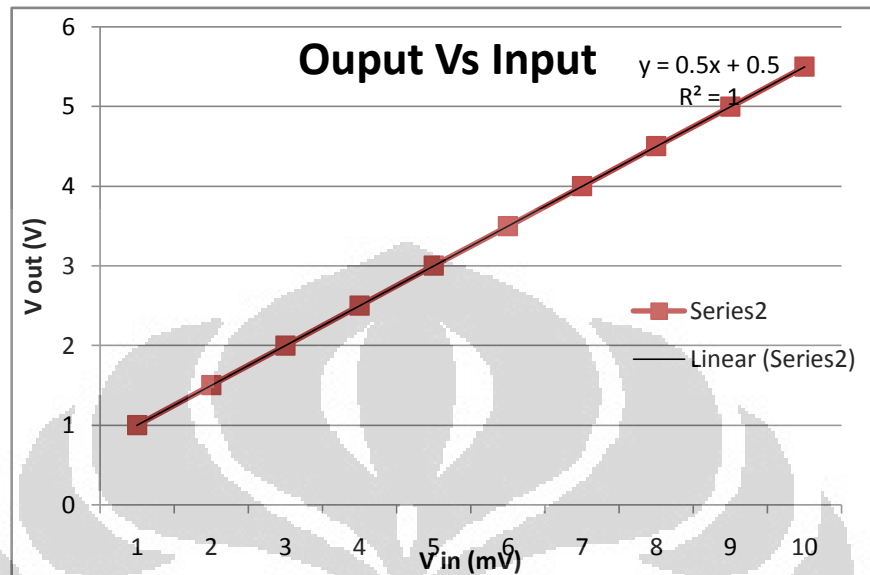
Mikrokontroler H8/3069F memiliki ADC internal 10 bit 8 kanal, namun yang dibutuhkan pada aplikasi ini hanyalah 1 kanal saja. ADC ini memiliki

jangkauan masukan dari 0 sampai 5 Volt. Pada aplikasi sistem logger ini, digunakan mode scan ADC. Hal ini bertujuan untuk mengantisipasi terjadinya error pada kanal tersebut dan juga membandingkan dengan kanal yang lain, disamping tujuan lainnya yaitu pengembangan penggunaan mikrokontroler H8/3069F pada aplikasi input berfluktuasi. Pengujian ADC internal mikrokontroler H8/3069F dilakukan dengan memberikan variasi input pada satu kanal yang kemudian melihat hasil konversinya pada hyperterminal PC.

Tabel 4.3 Tegangan *input* dan *output* pada ADC internal mikrokontroler H8/3069F

No	input (mV)	output (desimal)	output (V)
1	1	27	0.13
2	1.5	77	0.39
3	2	108	0.54
4	2.5	125	0.63
5	3	172	0.87
6	3.5	231	1.17
7	4	289	1.46
8	4.5	308	1.56
9	5	329	1.67
10	5.5	432	2.19
11	6	574	2.92
12	6.5	613	3.11
13	7	786	3.99
14	7.5	795	4.04
15	8	808	4.10
16	8.5	866	4.41
17	9	956	4.86

18	9.5	1007	5.12
19	10	1023	5.2



Gambar 4.3 Grafik respon output ADC internal mikrokontroler H8/3069

ADC internal mikrokontroler H8/3069F memiliki resolusi 10 bit. Hal ini berarti ADC tersebut dapat menghasilkan output 10 kode digital (10 bit = 1023), yang berarti ADC ini memiliki sensitivitas  $5.2/1023 \approx 0.0051$  V. Angka 5.2 ini adalah tegangan referensi yang dimiliki mikrokontroler H8 yang diukur dengan multimeter. Pada pengambilan data konversi, terlihat bahwa ketika kita memasukkan input tegangan maka tampilan pada hyperterminal PC adalah bilangan digital yang merupakan hasil konversi ke bilangan desimal. Apabila bilangan desimal tersebut kita ingin dikonversi ke dalam bilangan yang menyatakan tegangan maka kita harus melakukan perhitungan sebagai berikut :

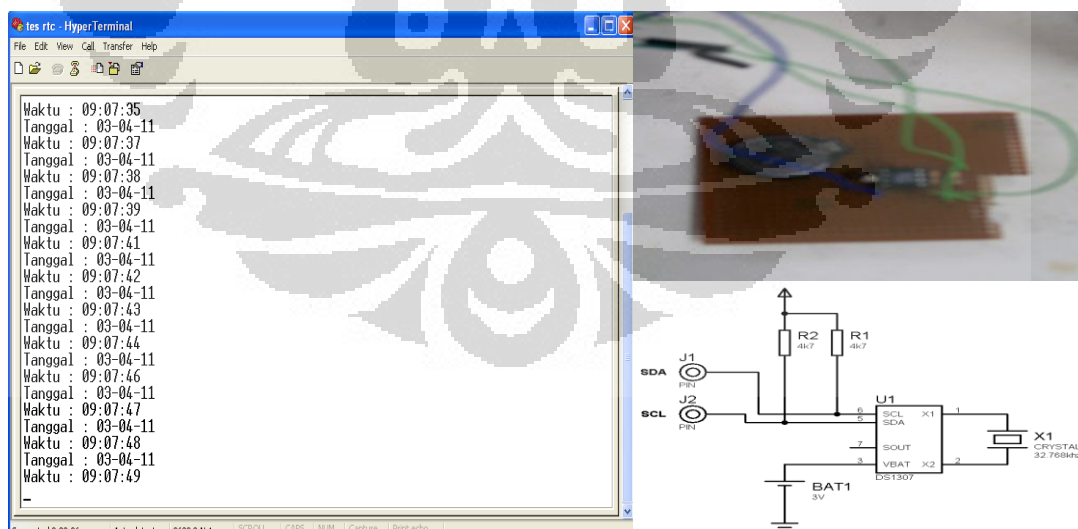
$$\text{Tegangan hasil konversi} = \frac{\text{output desimal}}{1023} \times 5,2 \text{ Volt}$$

Grafik diatas menunjukkan bahwa output yang dihasilkan cukup linear dengan input yang diberikan. Pada uji ADC ini, interval variasi input yang dapat diberikan hanya berkisar 1-10 mV, karena tegangan output desimalnya telah mencapai nilai maximum yaitu 1023, pada input 10 mV. Hal ini tidak sesuai

dengan tegangan input maksimum yang diharapkan, yaitu bisa mencapai 5V, sesuai dengan karakteristik ADC internal H8. Kesalahan ini sangat mungkin disebabkan oleh sumber tegangan input yang tidak stabil, dan karakteristik tegangan penguat instrumentasi IC AD620. Hal ini peneliti ketahui ketika peneliti membaca datasheet AD620 dan pengukuran tegangan input dengan voltmeter yang menunjukkan pembacaan tegangan yang berubah-ubah.

### 4.1.3 Uji Sistem Waktu

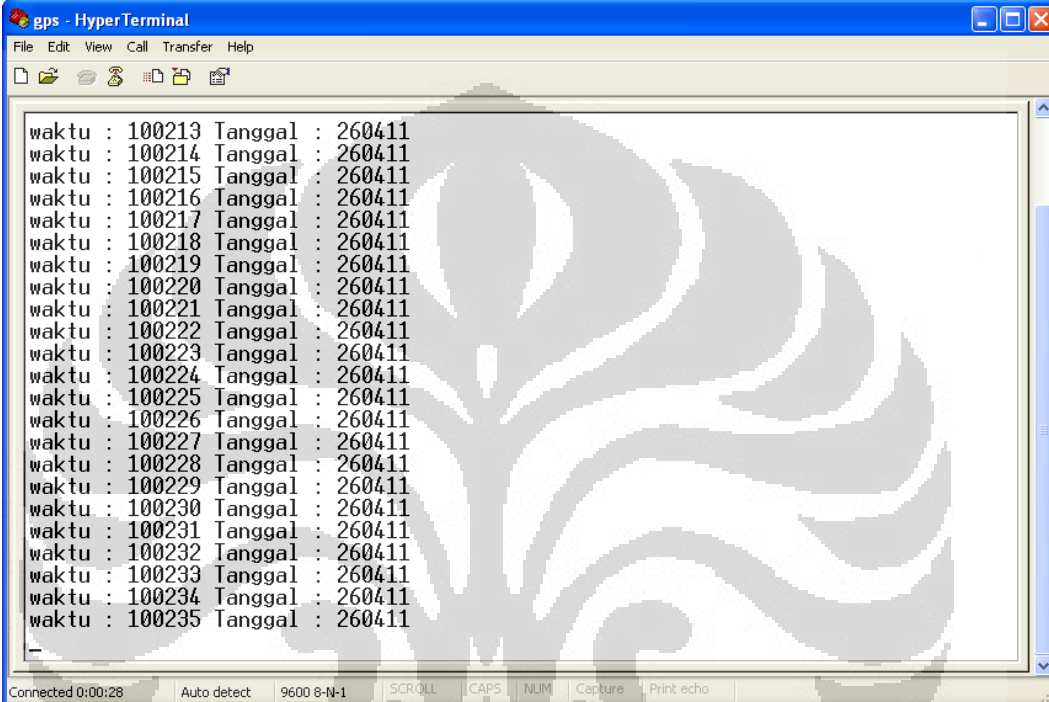
Uji sistem waktu ini, khususnya pengujian RTC, GPS, dan LCD bertujuan untuk mendapatkan informasi waktu yang akurat dan menunjukkan display sinkronisasi waktu sebelum dikombinasikan dengan data seismik hasil pengolahan mikrokontroler H8/3069F. Pengujian perangkat waktu ini dilakukan dengan menggunakan mikrokontroler H8/3069F dan melihat hasilnya pada hyperterminal PC. Pengujian RTC dimulai dengan pengaturan waktu secara manual oleh peneliti pada program yang telah dibuat. Pengaturan waktu manual ini hanya dilakukan sekali yang disesuaikan pada waktu pengambilan data, karena selanjutnya RTC akan mengcounting waktu sendiri secara otomatis. Berikut hasilnya



Gambar 4.4 RTC dan Hasil Output pada hyperterminal PC



Pengujian GPS dilakukan dengan mensetting konfigurasi NMEA 1803 oleh software GPS Gamin. Untuk mendapatkan waktu dan tanggal secara real-time digunakan sentence GPRMC dan parsing data agar yang terlihat pada hyperterminal PC hanya waktu dan tanggal saja.



```

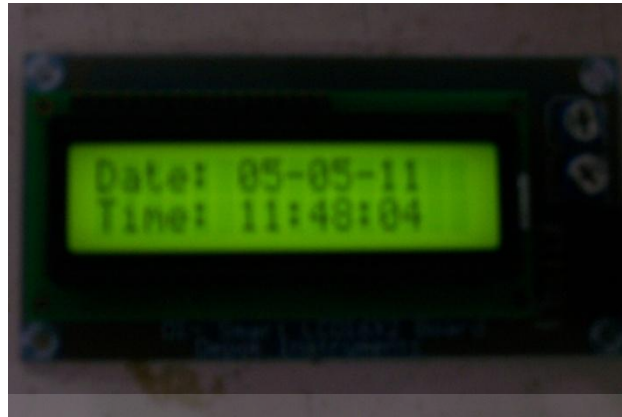
waktu : 100213 Tanggal : 260411
waktu : 100214 Tanggal : 260411
waktu : 100215 Tanggal : 260411
waktu : 100216 Tanggal : 260411
waktu : 100217 Tanggal : 260411
waktu : 100218 Tanggal : 260411
waktu : 100219 Tanggal : 260411
waktu : 100220 Tanggal : 260411
waktu : 100221 Tanggal : 260411
waktu : 100222 Tanggal : 260411
waktu : 100223 Tanggal : 260411
waktu : 100224 Tanggal : 260411
waktu : 100225 Tanggal : 260411
waktu : 100226 Tanggal : 260411
waktu : 100227 Tanggal : 260411
waktu : 100228 Tanggal : 260411
waktu : 100229 Tanggal : 260411
waktu : 100230 Tanggal : 260411
waktu : 100231 Tanggal : 260411
waktu : 100232 Tanggal : 260411
waktu : 100233 Tanggal : 260411
waktu : 100234 Tanggal : 260411
waktu : 100235 Tanggal : 260411

```

Gambar 4.5 Data waktu dan tanggal GPS setelah parsing

Waktu yang ditunjukkan pada gambar 4.6 bukanlah merupakan waktu untuk wilayah Indonesia, oleh sebab itu untuk mensinkronisasinya harus ditambahkan GMT +07.00.00 agar waktu sesuai dengan waktu di Indonesia.

Pengujian display LCD dilakukan dengan mengatur karakter dan delay pada pemrograman mikrokontroller H8/3069F. LCD ini memuat informasi tanggal dan waktu yang diambil dari hasil sinkronisasi GPS dan RTC.



Gambar 4.6 Display Waktu di LCD

## 4.2 PERANGKAT LUNAK

### 4.2.1 Pemrograman ADC Internal Mikrokontroler H8/3069F Mode Scan

Pemrograman ADC internal mikrokontroler H8/3069F yang digunakan pada logger data gelombang seismik ini adalah mode scan yang dapat mengkonversi delapan kanal input sekaligus. Namun, pada aplikasi ini hanya digunakan 1 kanal saja. Bahasa pemrograman yang digunakan adalah bahasa C yang di compile menjadi file.mot.

```

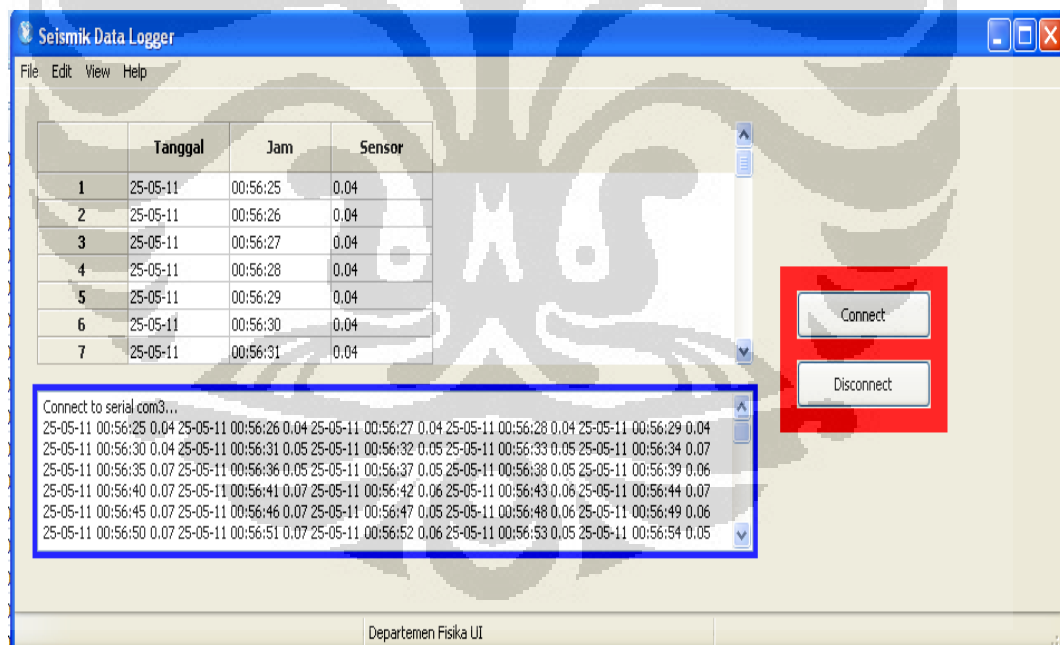
gsprtcadc - HyperTerminal
File Edit View Call Transfer Help
12-05-11 02:47:29 1023
12-05-11 02:47:30 1023
12-05-11 02:47:31 1023
12-05-11 02:47:32 1023
12-05-11 02:47:33 1023
12-05-11 02:47:34 1023
12-05-11 02:47:35 1023
12-05-11 02:47:34 1023
12-05-11 02:47:35 1023
12-05-11 02:47:36 1023
12-05-11 02:47:37 999
12-05-11 02:47:38 1023
12-05-11 02:47:39 1023
12-05-11 02:47:40 1023
12-05-11 02:47:41 1023
12-05-11 02:47:42 1023
12-05-11 02:47:43 1023
12-05-11 02:47:44 1023
12-05-11 02:47:45 1023
12-05-11 02:47:46 1023
12-05-11 02:47:47 1023
12-05-11 02:47:48 1023
12-05-11 02:47:49 1023
Connected 0:00:50 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

```

Gambar 4.7 Hasil konversi ADC (data seismik) pada hyperterminal dengan waktu GPS

#### 4.2.2 Pemograman GUI Phyton Seismik Data Logger

GUI (*Graphical User Interface*) pada sistem logger data gelombang seismik ini adalah berbasis bahasa pemrograman python. GUI ini digunakan untuk memudahkan *user* dalam pengambilan data gelombang seismik kemudian menyimpannya pada database. Fitur yang terdapat pada GUI ini adalah adanya 1 buah tombol kanal input (*connect*) yang merupakan fungsi untuk menentukan koneksi GUI . Selain itu, pada fitur ini juga kita dapat melihat secara *real time* gelombang masukan yang ditangkap. Data yang ditangkap ini kemudian disimpan ke dalam database MySQL secara otomatis. Data tersebut adalah berupa amplitudo beserta waktu gelombang.



Gambar 4.8 GUI Seismik Data Logger

	Tanggal	Jam	Data Geophone
<input type="checkbox"/>	25-05-11	00:56:25	0.04
<input type="checkbox"/>	25-05-11	00:56:26	0.04
<input type="checkbox"/>	25-05-11	00:56:27	0.04
<input type="checkbox"/>	25-05-11	00:56:28	0.04
<input type="checkbox"/>	25-05-11	00:56:29	0.04
<input type="checkbox"/>	25-05-11	00:56:30	0.04
<input type="checkbox"/>	25-05-11	00:56:31	0.05
<input type="checkbox"/>	25-05-11	00:56:32	0.05
<input type="checkbox"/>	25-05-11	00:56:33	0.05
<input type="checkbox"/>	25-05-11	00:56:34	0.07
<input type="checkbox"/>	25-05-11	00:56:35	0.07
<input type="checkbox"/>	25-05-11	00:56:36	0.05
<input type="checkbox"/>	25-05-11	00:56:37	0.05
<input type="checkbox"/>	25-05-11	00:56:38	0.05
<input type="checkbox"/>	25-05-11	00:56:39	0.06
<input type="checkbox"/>	25-05-11	00:56:40	0.07
<input type="checkbox"/>	25-05-11	00:56:41	0.07
<input type="checkbox"/>	25-05-11	00:56:42	0.06
<input type="checkbox"/>	25-05-11	00:56:43	0.06
<input type="checkbox"/>	25-05-11	00:56:44	0.07
<input type="checkbox"/>	25-05-11	00:56:45	0.07
<input type="checkbox"/>	25-05-11	00:56:46	0.07
<input type="checkbox"/>	25-05-11	00:56:47	0.05
<input type="checkbox"/>	25-05-11	00:56:48	0.06
<input type="checkbox"/>	25-05-11	00:56:49	0.06
<input type="checkbox"/>	25-05-11	00:56:50	0.07
<input type="checkbox"/>	25-05-11	00:56:51	0.07

Gambar 4.9 Database Seismik Data Logger

### 4.3 HASIL DATA LAPANGAN

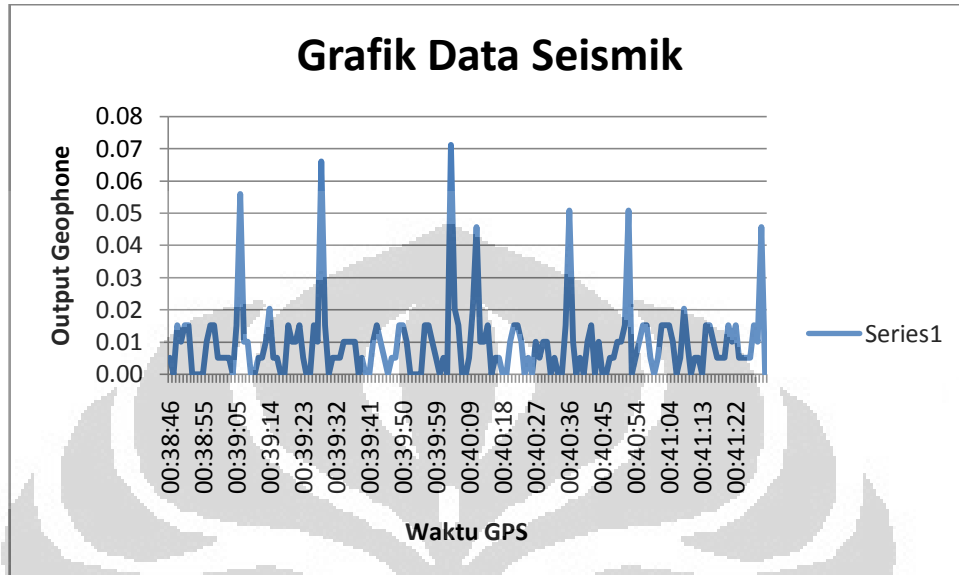
Hasil data lapangan ini diambil dengan menggunakan perangkat keras dan perangkat lunak yang sudah diujicobakan. Pengambilan data di lapangan dilakukan dengan menggunakan sumber getaran buatan yang berasal dari lempeng besi yang digetarkan dengan palu.



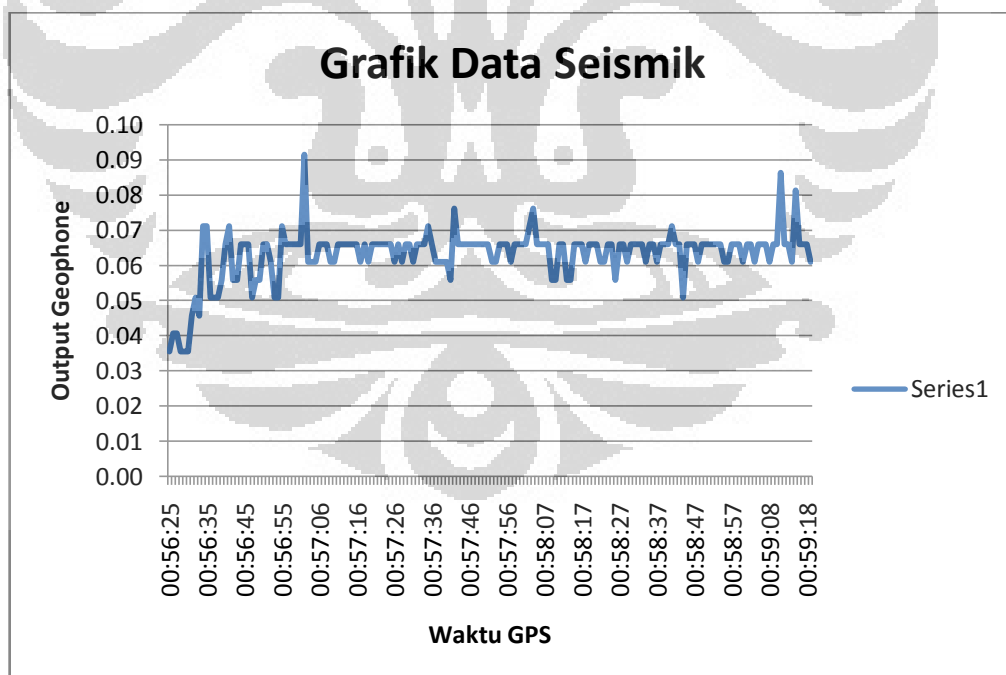
Gambar 4.10 Uji Sensor Geophone

Untuk menghasilkan data yang baik, tempat yang digunakan untuk pengambilan data adalah tempat yang memiliki struktur tanah yang solid agar perambatan gelombangnya baik dan jauh dari aktivitas masyarakat, karena

dikhawatirkan sumber getaran menjadi tercampur dan tidak murni. Data yang diambil divariasikan dengan jarak antara sensor geophon dengan sumber getaran masing-masing sejauh 2 meter dan 4 meter.



(a)



(b)

Gambar 4.11 (a) Grafik data seismik 2 meter. (b) Grafik data seismik 4 meter

Data seismik ini diambil dari database yang selanjutnya dibuat grafik seperti gambar di atas. Pada kedua grafik di atas terlihat bahwa output dari geophone sangat fluktuatif. Adanya kenyataan bahwa selama *logger* dilakukan, sinyal refleksi yang direkam mengalami perubahan bentuk atau terdistorsi selama penjarannya. Hal tersebut dapat dipengaruhi oleh banyak faktor, diantaranya sifat fisika batuan, geometri bawah permukaan, bentuk sumber, kinerja alat perekam, dan juga noise-noise alamiah.

Dalam grafik tersebut, untuk jarak sumber getaran dengan sensor sejauh 2 meter dan 4 meter, masing-masing diberikan pukulan ke sumber getar sebanyak 4 kali dan 2 kali. Waktu perekaman pada grafik menunjukkan waktu satelit GPS yang terhubung dan tersinkronisasi dengan waktu RTC. Dalam interval waktu yang terekam berdurasi sekitar 4 menit tersebut, kurang lebih bisa di asumsikan bahwa pemukulan dilakukan setiap 1 menit sekali dan 2 menit sekali. Idealnya, pengujian yang dilakukan haruslah 1 kali pukulan untuk sekali perekaman. Namun, hal itu bisa disiasati dengan melihat data yang ada pada database untuk hasil perekaman setiap menitnya, sehingga bisa di plot sesuai yang diinginkan.

## **BAB 5**

### **PENUTUP**

#### **5.1 KESIMPULAN**

- Hasil perekaman menunjukkan hubungan antara tegangan (sensor) dan waktu perambatan gelombang seismik
- Hasil perekaman seismik berhasil disimpan ke dalam *database MySQL* dan ditampilkan dalam bentuk dan GUI berbasis Phyton
- Hasil logger data belum dapat menggambarkan interpretasi bawah permukaan secara baik

#### **5.2 SARAN**

Penelitian lebih lanjut untuk yang dalam usaha pengembangan dan menambah kualitas penelitian ini dapat dilakukan dengan :

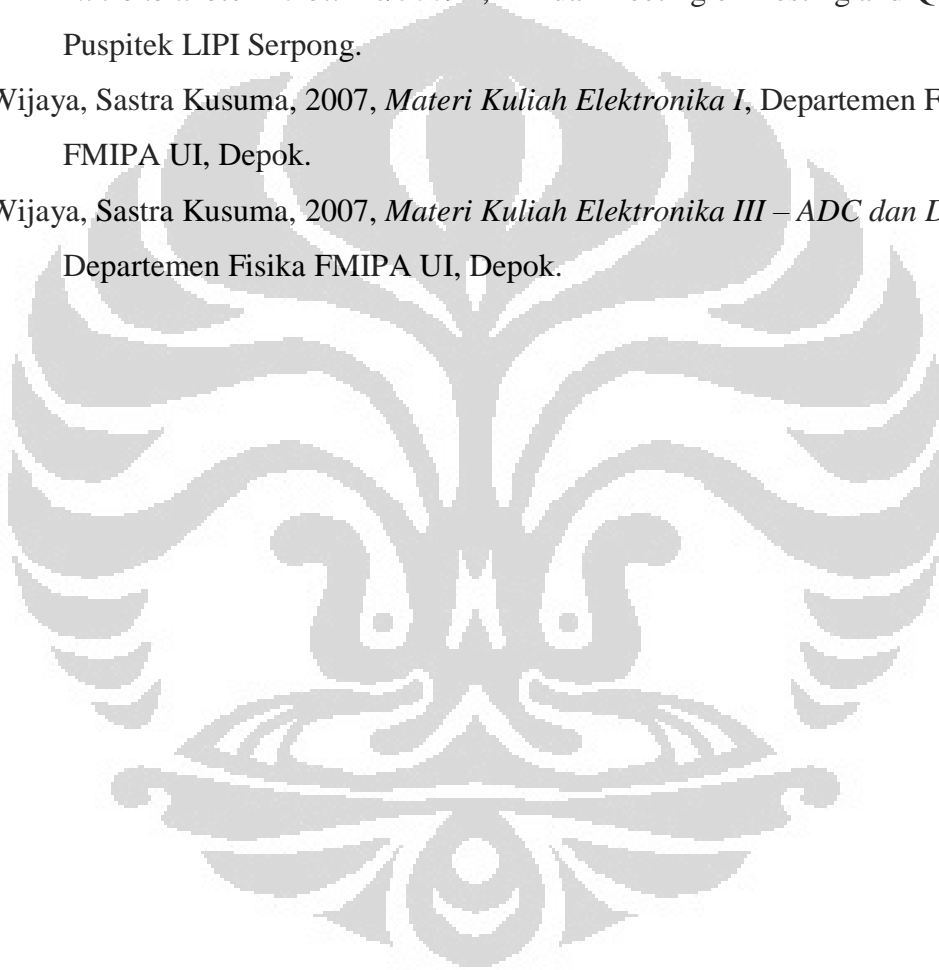
- Membuat sistem loger multichanel
- Membuat filter sinyal digital pada GUI Python

## DAFTAR ACUAN

- Alfa JK, MAS., et.al., 2008, *Osiloskop Digital Dua Kanal Menggunakan Mikrokontroller 16 Bit H8/3069F*, Seminar Nasional Sains dan Teknologi (SATEK) II, Lampung.
- Asrar, Allan, 2008, *Sistem Akuisisi Data Enam Channel Menggunakan Mikrokontroler H8/3069F*, Departemen Fisika FMIPA UI, Depok.
- Faulkenberry, Lucus M., 1982, *An introduction to Operational Amplifier 2nd edition*. John Wiley and sons, Canada.
- Gadallah, M. and Fisher R, 2009, *Exploration Geophysics: An Introduction*. : Springer, pp 5, German.
- Halliday, David and Robert Resnick, 1966, *Physics Combined Edition*, John Willey & Sons, Inc, New York.
- Hasan, Abu. 1999. *Sistem Akuisisi Data*. Institut Teknologi Bandung, Bandung.  
[http://www.renesas.com/fmwk.jsp?cnt=h8\\_family\\_landing.jsp&fp=/product/s/mpumcu/h8\\_family/](http://www.renesas.com/fmwk.jsp?cnt=h8_family_landing.jsp&fp=/product/s/mpumcu/h8_family/) pada tanggal 15 September 2009 pukul 15.27 WIB.
- Kirsch, R, 2006, *Groundwater Geophysics*. Springer, Berlin.
- Kurnadi, M. 2009. *Perancangan Sistem Akuisisi Data Gelombang Seismik Berbasis Mikrokontroler H8/3069F*. Departemen Fisika FMIPA UI, Depok
- Lowrie, W, 2007, *Fundamental of Geophysics*. Cambridge University Press, New York.
- Malvino, 1993. *Electronic Principle*. McGraw- Hill Book Company, New York.
- Norton Peter., et.al., 2005. *Beginning Python*. Wiley Publishing, Inc, Indiana.
- Nurhandy, Firdy Kusuma., et.al., 2008, *Alat Monitoring Tegangan RMS PLN Wilayah Kota Depok Berbasis Microcontroller*. Departemen Fisika FMIPA UI, Depok
- Purwanto, 2007, *Karakterisasi Reservoir Karbonat Dengan Menggunakan Analisis Avo dan Metode Lambda Mu Rho*, Jurusan Fisika FMIPA UGM, Yogyakarta.
- Putra, Muchtar Suhari., et.al., 2008, *Sistem kontrol inkubator melalui jaringan LAN menggunakan mikrokontroller H8/3069F 16-bit*, Annual Meeting on



- Testing and Quality, Puspitek LIPI Serpong.
- Renesas Technology Corp, 2009, *H8 Family Overview*, diakses dari
- Telford, W.M., et.al, 1999, *Applied Geophysics*. Cambridge University Press  
,Cambridge.
- Tim H8, 2007, *Modul Praktikum Renesas H8/3069F*, Departemen Fisika FMIPA  
UI, Depok.
- Wicaksono, Britantyo., et.al., 2007, *Sistem kontrol lengan robot menggunakan  
mikrokontroler 16-bit H8/3069F*, Annual Meeting on Testing and Quality,  
Puspitek LIPI Serpong.
- Wijaya, Sastra Kusuma, 2007, *Materi Kuliah Elektronika I*, Departemen Fisika,  
FMIPA UI, Depok.
- Wijaya, Sastra Kusuma, 2007, *Materi Kuliah Elektronika III – ADC dan DAC*.  
Departemen Fisika FMIPA UI, Depok.



## LAMPIRAN

Geopon 2 meter

Tanggal	Jam	Sensor
25-05-11	00:38:46	0.01
25-05-11	00:38:47	0.00
25-05-11	00:38:48	0.02
25-05-11	00:38:49	0.01
25-05-11	00:38:50	0.02
25-05-11	00:38:51	0.02
25-05-11	00:38:52	0.00
25-05-11	00:38:53	0.00
25-05-11	00:38:54	0.00
25-05-11	00:38:55	0.00
25-05-11	00:38:56	0.01
25-05-11	00:38:57	0.02
25-05-11	00:38:58	0.02
25-05-11	00:38:59	0.01
25-05-11	00:39:01	0.01
25-05-11	00:39:02	0.01
25-05-11	00:39:03	0.01
25-05-11	00:39:04	0.00
25-05-11	00:39:05	0.02
25-05-11	00:39:06	0.06
25-05-11	00:39:07	0.01
25-05-11	00:39:08	0.01
25-05-11	00:39:09	0.00
25-05-11	00:39:10	0.00
25-05-11	00:39:11	0.01
25-05-11	00:39:12	0.01
25-05-11	00:39:13	0.01
25-05-11	00:39:14	0.02
25-05-11	00:39:15	0.01
25-05-11	00:39:16	0.01
25-05-11	00:39:17	0.00
25-05-11	00:39:18	0.00
25-05-11	00:39:19	0.02
25-05-11	00:39:20	0.01
25-05-11	00:39:21	0.01
25-05-11	00:39:22	0.02
25-05-11	00:39:23	0.01

25-05-11	00:39:24	0.00
25-05-11	00:39:25	0.00
25-05-11	00:39:26	0.02
25-05-11	00:39:27	0.01
25-05-11	00:39:28	0.07
25-05-11	00:39:29	0.02
25-05-11	00:39:30	0.00
25-05-11	00:39:31	0.01
25-05-11	00:39:32	0.01
25-05-11	00:39:33	0.01
25-05-11	00:39:34	0.01
25-05-11	00:39:35	0.01
25-05-11	00:39:36	0.01
25-05-11	00:39:37	0.01
25-05-11	00:39:38	0.00
25-05-11	00:39:39	0.01
25-05-11	00:39:40	0.00
25-05-11	00:39:41	0.00
25-05-11	00:39:42	0.01
25-05-11	00:39:43	0.02
25-05-11	00:39:44	0.01
25-05-11	00:39:45	0.01
25-05-11	00:39:46	0.00
25-05-11	00:39:47	0.01
25-05-11	00:39:48	0.01
25-05-11	00:39:49	0.02
25-05-11	00:39:50	0.02
25-05-11	00:39:51	0.01
25-05-11	00:39:52	0.00
25-05-11	00:39:53	0.00
25-05-11	00:39:54	0.00
25-05-11	00:39:55	0.00
25-05-11	00:39:56	0.02
25-05-11	00:39:57	0.02
25-05-11	00:39:58	0.01
25-05-11	00:39:59	0.01
25-05-11	00:40:01	0.00
25-05-11	00:40:02	0.01
25-05-11	00:40:03	0.00

25-05-11	00:40:04	0.07
25-05-11	00:40:05	0.02
25-05-11	00:40:06	0.02
25-05-11	00:40:07	0.00
25-05-11	00:40:08	0.00
25-05-11	00:40:09	0.01
25-05-11	00:40:10	0.02
25-05-11	00:40:11	0.05
25-05-11	00:40:12	0.01
25-05-11	00:40:13	0.01
25-05-11	00:40:14	0.02
25-05-11	00:40:15	0.00
25-05-11	00:40:16	0.01
25-05-11	00:40:17	0.01
25-05-11	00:40:18	0.00
25-05-11	00:40:19	0.00
25-05-11	00:40:20	0.01
25-05-11	00:40:21	0.02
25-05-11	00:40:22	0.02
25-05-11	00:40:23	0.01
25-05-11	00:40:24	0.00
25-05-11	00:40:25	0.01
25-05-11	00:40:26	0.00
25-05-11	00:40:27	0.01
25-05-11	00:40:28	0.01
25-05-11	00:40:29	0.01
25-05-11	00:40:30	0.01
25-05-11	00:40:31	0.00
25-05-11	00:40:32	0.01
25-05-11	00:40:33	0.00
25-05-11	00:40:34	0.00
25-05-11	00:40:35	0.02
25-05-11	00:40:36	0.05
25-05-11	00:40:37	0.01
25-05-11	00:40:38	0.00
25-05-11	00:40:39	0.01
25-05-11	00:40:40	0.00
25-05-11	00:40:41	0.01
25-05-11	00:40:42	0.02
25-05-11	00:40:43	0.00
25-05-11	00:40:44	0.01

25-05-11	00:40:45	0.00
25-05-11	00:40:46	0.00
25-05-11	00:40:47	0.01
25-05-11	00:40:48	0.01
25-05-11	00:40:49	0.01
25-05-11	00:40:50	0.01
25-05-11	00:40:51	0.02
25-05-11	00:40:52	0.05
25-05-11	00:40:53	0.00
25-05-11	00:40:54	0.01
25-05-11	00:40:55	0.01
25-05-11	00:40:56	0.02
25-05-11	00:40:57	0.02
25-05-11	00:40:58	0.01
25-05-11	00:40:59	0.00
25-05-11	00:41:01	0.01
25-05-11	00:41:02	0.02
25-05-11	00:41:03	0.02
25-05-11	00:41:04	0.02
25-05-11	00:41:05	0.01
25-05-11	00:41:06	0.00
25-05-11	00:41:07	0.01
25-05-11	00:41:08	0.02
25-05-11	00:41:09	0.01
25-05-11	00:41:10	0.00
25-05-11	00:41:11	0.01
25-05-11	00:41:12	0.01
25-05-11	00:41:13	0.00
25-05-11	00:41:14	0.02
25-05-11	00:41:15	0.02
25-05-11	00:41:16	0.01
25-05-11	00:41:17	0.01
25-05-11	00:41:18	0.01
25-05-11	00:41:19	0.01
25-05-11	00:41:20	0.02
25-05-11	00:41:21	0.01
25-05-11	00:41:22	0.02
25-05-11	00:41:23	0.01
25-05-11	00:41:24	0.01
25-05-11	00:41:25	0.01
25-05-11	00:41:26	0.01

25-05-11	00:41:27	0.02
25-05-11	00:41:28	0.01

25-05-11	00:41:29	0.05
25-05-11	00:41:30	0.00

Geophone (4 meter)

Tanggal	Jam	Sensor
25-05-11	00:56:25	0.04
25-05-11	00:56:26	0.04
25-05-11	00:56:27	0.04
25-05-11	00:56:28	0.04
25-05-11	00:56:29	0.04
25-05-11	00:56:30	0.04
25-05-11	00:56:31	0.05
25-05-11	00:56:32	0.05
25-05-11	00:56:33	0.05
25-05-11	00:56:34	0.07
25-05-11	00:56:35	0.07
25-05-11	00:56:36	0.05
25-05-11	00:56:37	0.05
25-05-11	00:56:38	0.05
25-05-11	00:56:39	0.06
25-05-11	00:56:40	0.07
25-05-11	00:56:41	0.07
25-05-11	00:56:42	0.06
25-05-11	00:56:43	0.06
25-05-11	00:56:44	0.07
25-05-11	00:56:45	0.07
25-05-11	00:56:46	0.07
25-05-11	00:56:47	0.05
25-05-11	00:56:48	0.06
25-05-11	00:56:49	0.06
25-05-11	00:56:50	0.07
25-05-11	00:56:51	0.07
25-05-11	00:56:52	0.06
25-05-11	00:56:53	0.05
25-05-11	00:56:54	0.05
25-05-11	00:56:55	0.07
25-05-11	00:56:56	0.07
25-05-11	00:56:57	0.07
25-05-11	00:56:58	0.07
25-05-11	00:56:59	0.07

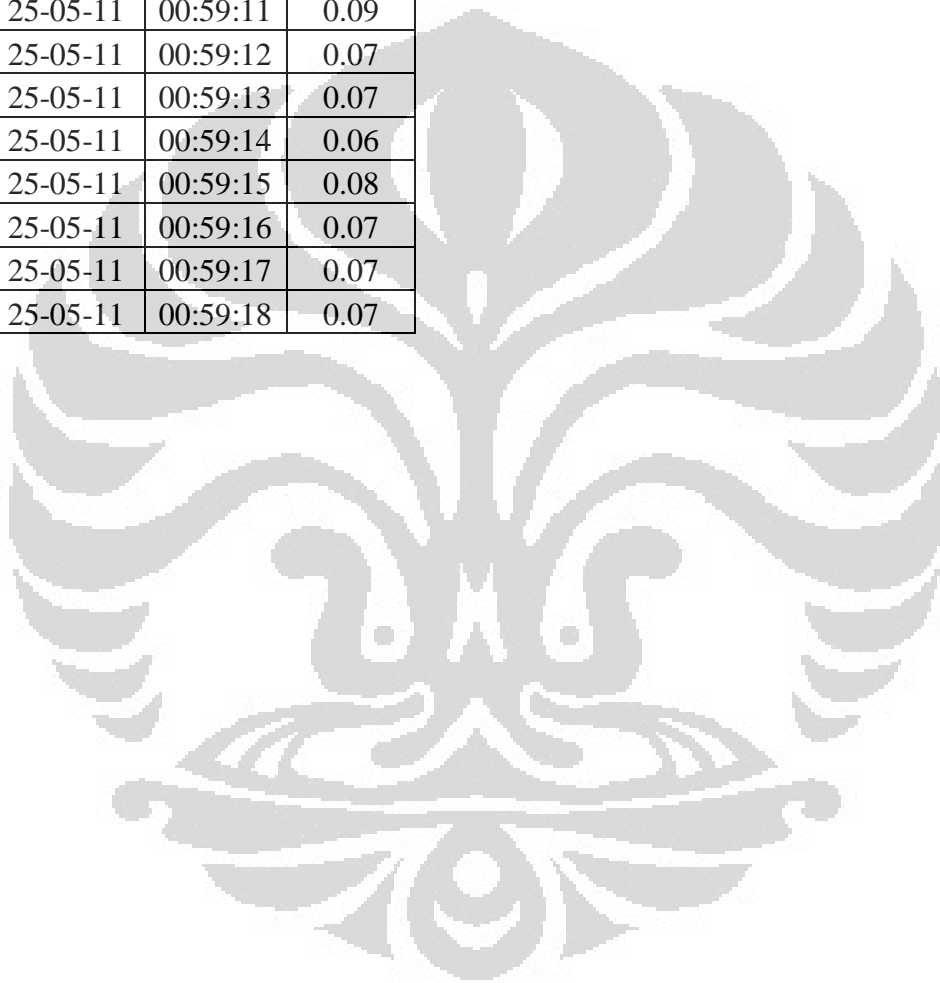
25-05-11	00:57:01	0.07
25-05-11	00:57:02	0.09
25-05-11	00:57:03	0.06
25-05-11	00:57:04	0.06
25-05-11	00:57:05	0.06
25-05-11	00:57:06	0.07
25-05-11	00:57:07	0.07
25-05-11	00:57:08	0.07
25-05-11	00:57:09	0.06
25-05-11	00:57:10	0.06
25-05-11	00:57:11	0.07
25-05-11	00:57:12	0.07
25-05-11	00:57:13	0.07
25-05-11	00:57:14	0.07
25-05-11	00:57:15	0.07
25-05-11	00:57:16	0.07
25-05-11	00:57:17	0.06
25-05-11	00:57:18	0.07
25-05-11	00:57:19	0.06
25-05-11	00:57:20	0.07
25-05-11	00:57:21	0.07
25-05-11	00:57:22	0.07
25-05-11	00:57:23	0.07
25-05-11	00:57:24	0.07
25-05-11	00:57:25	0.07
25-05-11	00:57:26	0.06
25-05-11	00:57:27	0.07
25-05-11	00:57:28	0.06
25-05-11	00:57:29	0.07
25-05-11	00:57:30	0.07
25-05-11	00:57:31	0.06
25-05-11	00:57:32	0.07
25-05-11	00:57:33	0.07
25-05-11	00:57:34	0.07
25-05-11	00:57:35	0.07
25-05-11	00:57:36	0.07
25-05-11	00:57:37	0.06

25-05-11	00:57:38	0.06
25-05-11	00:57:39	0.06
25-05-11	00:57:40	0.06
25-05-11	00:57:41	0.06
25-05-11	00:57:42	0.08
25-05-11	00:57:43	0.07
25-05-11	00:57:44	0.07
25-05-11	00:57:45	0.07
25-05-11	00:57:46	0.07
25-05-11	00:57:47	0.07
25-05-11	00:57:48	0.07
25-05-11	00:57:49	0.07
25-05-11	00:57:50	0.07
25-05-11	00:57:51	0.07
25-05-11	00:57:52	0.06
25-05-11	00:57:53	0.06
25-05-11	00:57:54	0.07
25-05-11	00:57:55	0.07
25-05-11	00:57:56	0.07
25-05-11	00:57:57	0.06
25-05-11	00:57:58	0.07
25-05-11	00:57:59	0.07
25-05-11	00:58:01	0.07
25-05-11	00:58:02	0.07
25-05-11	00:58:03	0.07
25-05-11	00:58:04	0.08
25-05-11	00:58:05	0.07
25-05-11	00:58:06	0.07
25-05-11	00:58:07	0.07
25-05-11	00:58:08	0.07
25-05-11	00:58:09	0.06
25-05-11	00:58:10	0.06
25-05-11	00:58:11	0.07
25-05-11	00:58:12	0.07
25-05-11	00:58:13	0.06
25-05-11	00:58:14	0.06
25-05-11	00:58:15	0.07
25-05-11	00:58:16	0.07
25-05-11	00:58:17	0.07
25-05-11	00:58:18	0.06
25-05-11	00:58:19	0.07

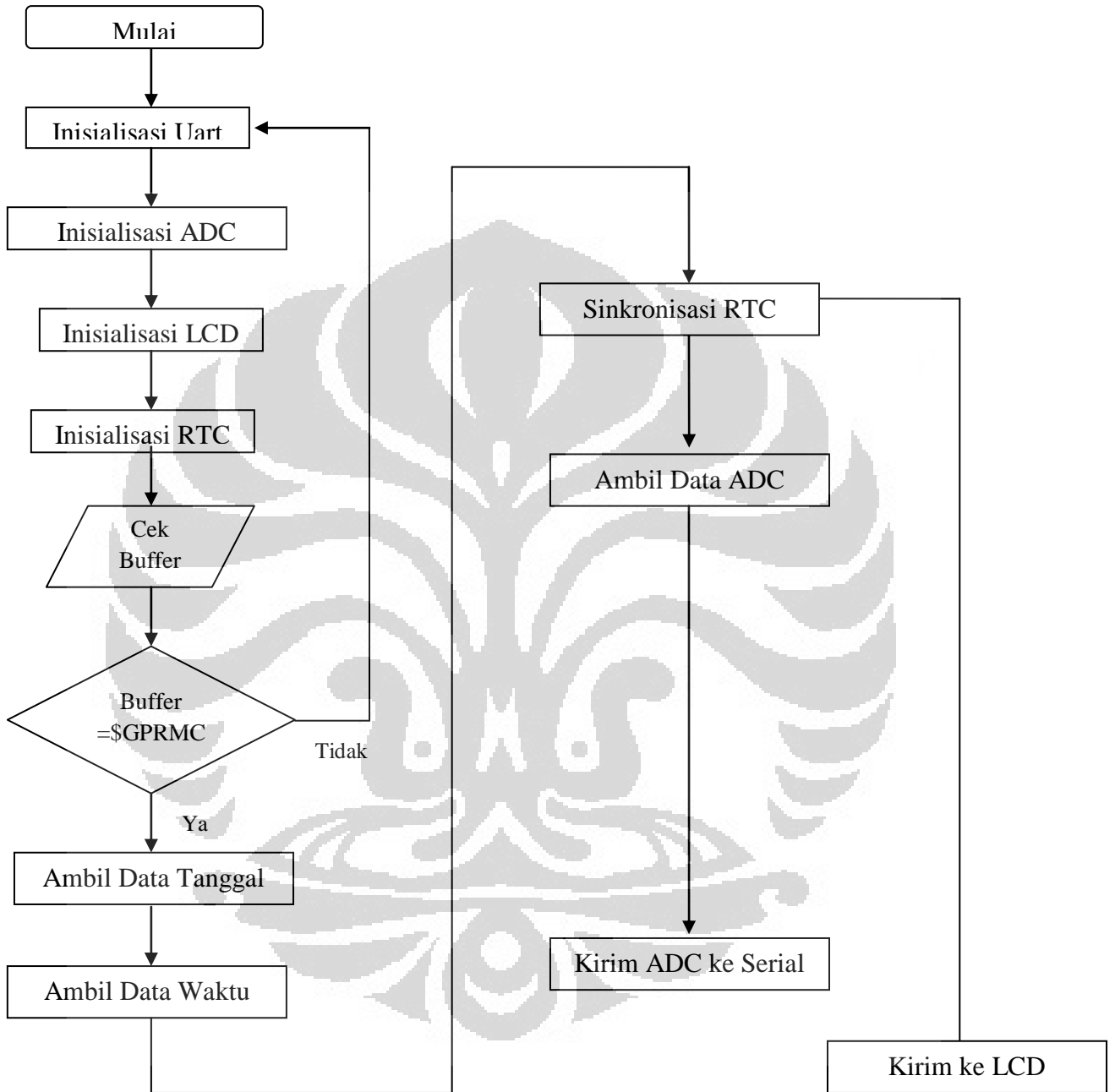
25-05-11	00:58:20	0.07
25-05-11	00:58:21	0.07
25-05-11	00:58:22	0.06
25-05-11	00:58:23	0.06
25-05-11	00:58:24	0.07
25-05-11	00:58:25	0.07
25-05-11	00:58:26	0.06
25-05-11	00:58:27	0.07
25-05-11	00:58:28	0.07
25-05-11	00:58:29	0.06
25-05-11	00:58:30	0.07
25-05-11	00:58:31	0.07
25-05-11	00:58:32	0.07
25-05-11	00:58:33	0.07
25-05-11	00:58:34	0.06
25-05-11	00:58:35	0.07
25-05-11	00:58:36	0.07
25-05-11	00:58:37	0.06
25-05-11	00:58:38	0.07
25-05-11	00:58:39	0.07
25-05-11	00:58:40	0.07
25-05-11	00:58:41	0.07
25-05-11	00:58:42	0.07
25-05-11	00:58:43	0.07
25-05-11	00:58:44	0.05
25-05-11	00:58:45	0.07
25-05-11	00:58:46	0.07
25-05-11	00:58:47	0.07
25-05-11	00:58:48	0.06
25-05-11	00:58:49	0.07
25-05-11	00:58:50	0.07
25-05-11	00:58:51	0.07
25-05-11	00:58:52	0.07
25-05-11	00:58:53	0.07
25-05-11	00:58:54	0.07
25-05-11	00:58:55	0.06
25-05-11	00:58:56	0.06
25-05-11	00:58:57	0.07
25-05-11	00:58:58	0.07
25-05-11	00:58:59	0.07
25-05-11	00:59:01	0.06

25-05-11	00:59:02	0.07
25-05-11	00:59:03	0.07
25-05-11	00:59:04	0.06
25-05-11	00:59:05	0.07
25-05-11	00:59:06	0.07
25-05-11	00:59:07	0.07
25-05-11	00:59:08	0.06
25-05-11	00:59:09	0.07
25-05-11	00:59:10	0.07
25-05-11	00:59:11	0.09
25-05-11	00:59:12	0.07
25-05-11	00:59:13	0.07
25-05-11	00:59:14	0.06
25-05-11	00:59:15	0.08
25-05-11	00:59:16	0.07
25-05-11	00:59:17	0.07
25-05-11	00:59:18	0.07

25-05-11	00:59:19	0.06
----------	----------	------



Flowchart program seismik data logger



```
/*Program Utama Seismik Data Logger Tersinkronisasi GPS*/
```

```
#include "mydef.h"  
#include "uart.h"  
#include "inthandler.h"  
#include "io306x.h"  
//#include "ioh83069.h"  
#include "uart.h"  
#include "rtc.h"  
#include "delay.h"  
#include "interrupt.h"  
#include "romcode.h"  
#include "adc.h"  
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#define DATA      P1DR.BYTE  
#define DATAS     P1DDR  
#define LCD_HOME  1  
#define LCD_RS    P1DR.BIT.B0  
#define LCD_E     P1DR.BIT.B2  
#define LCD_D4    P1DR.BIT.B4  
#define LCD_D5    P1DR.BIT.B5  
#define LCD_D6    P1DR.BIT.B6  
#define LCD_D7    P1DR.BIT.B7
```

```
#define NSENSOR 1  
char cmdbuf[512], data, waktu[10], tanggal[10];  
unsigned short i;  
unsigned short cmdln;
```

```
//char btime[10], bdate[10];  
unsigned short adcbuf[10];  
char buf[50];
```

```
//unsigned char i;
```

```
void lcd_send_char(unsigned char data)  
{
```

```
    DATA = (data & 0xF0); //Tulis nible atas  
    DATA |= 1 << LCD_RS; //Kirim data  
    DATA |= 1 << LCD_E; //Enable  
    msdelay(1);  
    DATA |= 0 << LCD_E; //Disable  
    msdelay(1);
```

```
    DATA = ((data & 0x0F) << 4); //Tulis nible bawah  
    DATA |= 1 << LCD_RS; //Kirim data  
    DATA |= 1 << LCD_E; //Enable  
    msdelay(1);  
    DATA |= 0 << LCD_E; //Disable  
    msdelay(1);
```

```
    DATA = (data & 0xF0); //Tulis nible atas  
    DATA |= 0x01; //Kirim data  
    DATA |= 0x04; //Enable  
    msdelay(2);  
    DATA |= 0x00; //Disable
```



```

msdelay(2);

DATA = ((data & 0x0F) << 4); //Tulis nibble bawah
DATA |= 0x01; //Kirim data
DATA |= 0x04; //Enable
msdelay(2);
DATA |= 0x00; //Disable
msdelay(2);
}

```

```

void lcd_send_command(unsigned char cmd)
{
    /*
    DATA = (cmd & 0XF0); //Tulis nibble atas
    DATA |= 0 << LCD_RS; //Tulis command
    DATA |= 1 << LCD_E; //Enable
    msdelay(1);
    DATA |= 0 << LCD_E;
    msdelay(1);

    DATA = ((cmd & 0X0F) << 4); //Tulis nibble bawah
    DATA |= 0 << LCD_RS; //Tulis command
    DATA |= 1 << LCD_E; //Enable
    msdelay(1);
    DATA |= 0 << LCD_E;
    msdelay(1);
    */

    DATA = (cmd & 0XF0); //Tulis nibble atas
    DATA |= 0x00; //Tulis command
    DATA |= 0x04; //Enable
    msdelay(2);
    DATA |= 0x00;
    msdelay(2);

    DATA = ((cmd & 0X0F) << 4); //Tulis nibble bawah
    DATA |= 0x00; //Tulis command
    DATA |= 0x04; //Enable
    msdelay(2);
    DATA |= 0x00;
    msdelay(2);
}

```

```

void lcd_clear(void)
{
    lcd_send_command(0x01); //Clear display
    msdelay(2);
}

```

```

void lcd_lf(void)
{
    lcd_send_command(0xC0);
    msdelay(2);
}

```

```

void lcd_init(void)
{
    msdelay(15);
    DATAS = 0xFF;
}

```

```

DATA |= 0xF0;
DATA |= 0x05;
//DATA |= 1 << LCD_D7 | 1 << LCD_D6 | 1 << LCD_D5 | 1 << LCD_D4;
//DATA |= 1 << LCD_E | 1 << LCD_RS;

lcd_send_command(0x28); //Mode 4-bit 2 line 4x8
lcd_send_command(0x0C); //display on, cursor off, bling off)
lcd_send_command(0x06); //shift inc, invisible
}

void lcd_printc(unsigned char data)
{
    if (data == '\n'){
        lcd_lf();
    }
    else {
        lcd_send_char(data);
    }
}

void lcd_prints(unsigned char *data)
{
    while (*data != '\0')
    {
        lcd_printc(*data);
        data++;
    }
}

void lcd_string(char* str)
{
    unsigned char x=0;
    while (str[x] != 0)
    {
        msdelay(1);
        lcd_send_char(str[x]);
        x++;
    }
}

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) {
            //uart_ch(data);
        }
    }
}

```

```

    cmdbuf[cmdln] = data;
    cmdln++;
} else {
    //uart_strP("\r\n");
    cmdbuf[cmdln] = '\0';

    if (cmdbuf[0] != '$') {
        cmdln = 0;
    } else {
        if ((cmdbuf[1] == 'G') && (cmdbuf[2] == 'P') &&
(cmdbuf[3] == 'R') && (cmdbuf[4] == 'M') && (cmdbuf[5] == 'C')) {
            //uart_str(&cmdbuf[0]);
            //Parsing waktu
            //waktu mulai pada posisi ke 7 dengan panjang 6
            //for (i = 0; i < 6; i++) {
            waktu[0] = cmdbuf[7];
            waktu[1] = cmdbuf[8];
            waktu[2] = ':';
            waktu[3] = cmdbuf[9];
            waktu[4] = cmdbuf[10];
            waktu[5] = ':';
            waktu[6] = cmdbuf[11];
            waktu[7] = cmdbuf[12];
            waktu[8] = '\0';
            //}
            //i++;
            //delay_ms(10800000);
            //Tampilkan diuart
            //uart_str("waktu : ");

            settime(&waktu[0]);
            getrtc(&tanggal[0], &waktu[0]);
            lcd_send_command(LCD_HOME);
            lcd_send_command(0x80+0);
            lcd_prints("time: ");
            //lcd_lf();
            lcd_prints(&waktu[0]);
            uart_str(&waktu[0]);

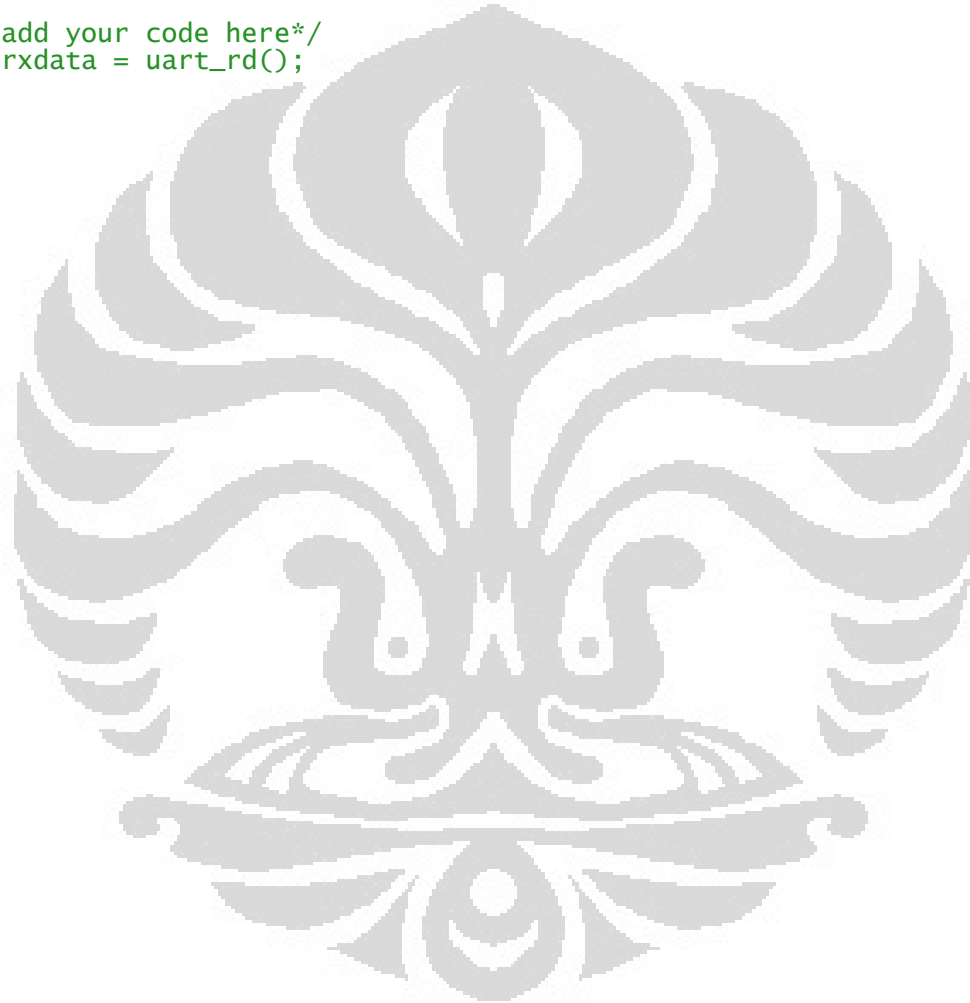
            uart_strP(" ");

            //for (i = 0; i < 6; i++) {
            tanggal[0] = cmdbuf[43];
            tanggal[1] = cmdbuf[44];
            tanggal[2] = '-';
            tanggal[3] = cmdbuf[45];
            tanggal[4] = cmdbuf[46];
            tanggal[5] = '-';
            tanggal[6] = cmdbuf[47];
            tanggal[7] = cmdbuf[48];
            tanggal[8] = '\0';
            //}
            //i++;

```



```
lcd_init();  
msdelay(100);  
  
while(1);  
return 0;  
}  
  
//Interrupt handler  
void INT_RXI1(void)  
{  
    /*add your code here*/  
    //rxdata = uart_rd();  
}
```



## Main Program GUI Seismik Datan Logger

```
import wx, wx.grid
import os, time
import serial, threading
import MySQLdb as sql
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 = ['Tanggal', 'Jam', 'Sensor']
num_rows = 40
num_cols = 3

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, self.eventType, windowID)
        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='Seismik Data Logger', size=(610,335))
        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_tanggal = []
self.tbl_jam = []
self.tbl_sensor = []

```

```

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

```

```

#----- panel-panel utama -----
pn1 = wx.Panel(self, -1)
#pn1.SetBackgroundColour('medium forest green')
pn2 = wx.Panel(self, -1)
pn2.SetBackgroundColour('blue')
#pn3 = wx.Panel(self, -1)
#pn3.SetBackgroundColour('medium forest green')
#pn4 = wx.Panel(self, -1)
#pn4.SetBackgroundColour('grey')
pn5 = wx.Panel(self, -1)
pn5.SetBackgroundColour('red')

#----- widget panel 1-----
self.grid = wx.grid.Grid(pn1, -1, size=(642,151))
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)
pn1.SetSizer(border1)

#----- widget panel 2-----
self.log = wx.TextCtrl(pn2, -1, "", size=(642, 100), style = wx.TE_MULTILINE |
wx.TE_PROCESS_ENTER)
sizer2 = wx.BoxSizer(wx.VERTICAL)
sizer2.Add(self.log, 1, wx.EXPAND | wx.ALL, 4)
pn2.SetSizer(sizer2)

#----- widget panel 3 -----

#----- grafik-----

#----- widget panel 4 -----

#----- widget panel 5 -----

```

```

btn1 = wx.Button(pnl5, -1, 'Connect', pos = (225, 5), size = (120, 30))
btn2 = wx.Button(pnl5, -1, 'Disconnect', pos = (315, 5), size = (120, 30))
#btn3 = wx.Button(pnl5, -1, 'Convert', size = (120, 30))
#btn4 = wx.Button(pnl5, -1, 'Show Data', size = (120, 30))

```

```

self.Bind(wx.EVT_BUTTON, self.OnBtn1, btn1)
self.Bind(wx.EVT_BUTTON, self.OnBtn2, btn2)
#self.Bind(wx.EVT_BUTTON, self.OnBtn3, btn3)
#self.Bind(wx.EVT_BUTTON, self.OnBtn4, btn4)

```

```

bbtn = wx.GridSizer(1, 1, 3, 3)
bbtn.Add(btn1, 1, wx.EXPAND | wx.ALL, 5)
bbtn.Add(btn2, 1, wx.EXPAND | wx.ALL, 5)
#bbtn.Add(btn3, 2, wx.EXPAND | wx.ALL, 10)
#bbtn.Add(btn4, 2, wx.EXPAND | wx.ALL, 10)

```

```

sizer4 = wx.BoxSizer(wx.VERTICAL)
sizer4.Add(bbtn, 2, wx.EXPAND | wx.ALL, 10)

```

```

sizer5 = wx.BoxSizer()
sizer5.Add(sizer4, 1, wx.EXPAND | wx.ALL, 0)
pnl5.SetSizer(sizer5)

```

```

#-----penyatuan panel-----

```

```

box = wx.BoxSizer(wx.VERTICAL)
#box.Add(pnl1, 0, wx.ALIGN_CENTER_HORIZONTAL | wx.ALL, 0)
box.Add(pnl1, 0, wx.ALIGN_CENTER_VERTICAL | wx.ALL, 4)
box.Add(pnl2, 0, wx.ALIGN_CENTER_VERTICAL | wx.ALL, 4)
#box.Add(pnl3, 1, wx.EXPAND | wx.ALL, 4)

```

```

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

```

```

box3 = wx.BoxSizer(wx.VERTICAL)
#box3.Add(pnl4, 0, wx.ALIGN_CENTER | wx.ALL, 4)
box3.Add(pnl5, 0, wx.LEFT | wx.ALL, 4)

```

```

box4 = wx.BoxSizer(wx.VERTICAL)
box4.Add(box3, 1, wx.ALIGN_CENTER | wx.ALL, 4)

```

```

box5 = wx.BoxSizer(wx.HORIZONTAL)
box5.Add(box2, 0, wx.EXPAND | wx.ALL, 4)
box5.Add(box4, 0, wx.ALIGN_CENTER | wx.ALL, 0)

```

```

box6 = wx.BoxSizer()
box6.Add(box5, 1, wx.EXPAND | wx.ALL, 4)
self.SetSizer(box6)
self.Layout()

```

```

#-----menu bar-----

```

```

menuBar = wx.MenuBar()

```

```

mfile = wx.Menu() #Isi dari mfile
self.saveMenuItem = mfile.Append(-1, '&Save', 'Save')
self.printMenuItem = 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 = mhhelp.Append(-1, "&Help\tF1", "Help")
mhhelp.AppendSeparator()
self.aboutMenuItem = mhhelp.Append(-1, "&About", "About")

#Definisi dari Menubar
menuBar.Append(mfile, '&File')
menuBar.Append(medit, '&Edit')
menuBar.Append(mview, '&View')
menuBar.Append(mhhelp, '&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)

#=====menampilkan waktu=====

self.jam = wx.StaticText(pnl1, -1, "hh:mm:ss")
self.Timer = wx.Timer(self)
self.Bind(wx.EVT_TIMER, self.OnTimer, self.Timer)
self.Timer.Start(1000)

# layout utamaSizer
vbox = wx.BoxSizer(wx.VERTICAL)
vbox.Add(btnsizer, 0, wx.EXPAND | wx.ALL, 0)
vbox.Add(sizer, 0, wx.EXPAND | wx.ALL, 0)

border = wx.BoxSizer()
border.Add(vbox, 1, wx.EXPAND | wx.ALL, 0)
pnl1.SetSizer(border)
pnl1.Layout()

def Waktu(self):
    t = time.localtime(time.time())
    st = time.strftime("%A, %d-%b-%Y %I:%M:%S",t)
    self.statusbar.SetStatusText(st, 0) #0 = u/letak di sebelah kiri
    self.jam.SetLabel(st)

def OnTimer(self, event):
    self.Waktu()

```

```

#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_tanggal = ""
    data_jam = ""
    data_sensor = ""

    self.data.append(text)

    if self.data[-1][-1] == '\n' and len(self.data) > 3:
        temp = ".join(self.data)
        data_mentah = temp.split(' ')
        #self.data.append(text)
        self.data = []

        if data_mentah > 3:
            data_tanggal = data_mentah[0]
            data_jam = data_mentah[1]
            data_sensor = ((float(str(data_mentah[2]))*5.2)) / 1023

```

```

self.tbl_tanggal.append(data_mentah[0])
self.tbl_jam.append(data_mentah[1])
#self.tbl_tekanan.append(data_mentah[5])
self.tbl_sensor.append(data_sensor)

for i in range(num_rows):
    try:
        self.grid.SetCellValue(i, 0, '%s' % self.tbl_tanggal[-(i+1)])
        self.grid.SetCellValue(i, 1, '%s' % self.tbl_jam[-(i+1)])
        self.grid.SetCellValue(i, 2, '%.2f' % self.tbl_sensor[-(i+1)])
        #self.grid.SetCellValue(i, 5, '%s' % self.tbl_tekanan[-(i+1)])
    except IndexError:
        pass

#print data_mentah #self.data

self.log.WriteText(text)

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()

#def OnBtn3(self, event):
#    self.log.WriteText('Konversi Untuk Sensor DS18B20...\n')
#    self.Write('%c')
#    self.ser.write(RETURN)

#def OnBtn4(self, event):
#    self.log.WriteText('Membaca Sensor DS18B20...\n')
#    self.ser.Write('%s0')
#    self.ser.write(RETURN)

#def OnConnect(self):
#    self.log.WriteText('> Connect to serial '+MYPOR+'...\n')

#try:
#    self.ser = serial.Serial(MYPOR)
#    self.StartThread()

```

```

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

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

def WriteCommand(self, command):
    self.ser.write(command)
    self.ser.write(RETURN)
    #self.ser.write('$c'+RETURN+CARRIAGE)

def OnRun(self, event):
    self.OnConnect() #Untuk melakukan koneksi ke serial
    self.WriteCommand('$sa') #Untuk memberikan command melalui serial

def OnStop(self, event):
    #self.OnDisconnect()
    self.WriteCommand('$so')

def InsertData(stanggal, sjam, slatitude, slongitude, stemperatur, stekanan):
    state = "INSERT INTO tbl_data (tanggal,jam,sensor) VALUES \
('"+ stanggal +"', '"+ sjam +"', '"+ ssensor +"');"

    conn = sql.connect(user=pUser, passwd=pPasswd, host=pHost, db=pDb)
    cur = conn.cursor()
    cur.execute(state)
    cur.close()
    conn.commit()
    conn.close()

def DataNext1(self):
    return self.tbl_sensor

#def DataNext2(self):
    #return self.tbl_tekanan

def OnRedrawTimer(self):
    self.tbl_sensor.append(self.DataNext1())
    #self.tbl_tekanan.append(self.DataNext2())
    self.draw_plot()

```