

LAPORAN TUGAS AKHIR

**APLIKASI JARINGAN SENSOR NIRKABEL UNTUK PENGUKURAN
PERGERAKAN TANAH MENGGUNAKAN SENSOR PERCEPATAN 3 SUMBU
PADA DAERAH RAWAN LONGSOR**



Oleh :

DWI KURNIAWAN

I1A006019

**KEMENTERIAN PENDIDIKAN NASIONAL
UNIVERSITAS JENDERAL SOEDIRMAN
FAKULTAS SAINS DAN TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
PURWOKERTO
2013**

LEMBAR PENGESAHAN

TUGAS AKHIR

APLIKASI JARINGAN SENSOR NIRKABEL UNTUK PENGUKURAN PERGERAKAN TANAH MENGGUNAKAN SENSOR PERCEPATAN 3 SUMBU PADA DAERAH RAWAN LONGSOR

Diajukan Sebagai Syarat Untuk Memperoleh Gelar Sarjana Teknik
Pada Fakultas Sains Dan Teknik Program Studi Teknik Elektro
Universitas Jenderal Soedirman

Oleh:

DWI KURNIAWAN

I1A006019

Diterima dan disetujui

Pada tanggal

Dosen Pembimbing I

Dosen Pembimbing II

Imron Rosyadi,ST.,MSc
NIP. 197909242003121003

Azis Wisnu Widhi N,ST.,M.Eng
NIP. 197811022003121002

Mengetahui,
Ketua Jurusan Teknik

Hari Prasetijo, ST., MT.
NIP. 197308222000121001

TENTANG PENULIS



DWI KURNIAWAN, lahir di Cilacap Jawa Tengah pada tanggal 27 November 1987 yang merupakan putra kedua dari pasangan MUNARTO dan RAWEN.

Penulis menempuh pendidikan dasar di SD Pasuruhan II Kec. Binangun kemudian melanjutkan ke jenjang SLTP di SLTP N1 Binangun dan menempuh pendidikan menengah di SMA N1 Binangun kab. Cilacap. Semenjak dini penulis telah tertarik terhadap dunia elektero dan semasa menempuh pendidikan dasar penulis pernah ditanya cita-citanya dan menjawab ingin menjadi insinyur seperti pak HABIBIE. Mungkin karena latar belakang itulah akhirnya penulis melanjutkan ke jenjang perguruan tinggi dan memilih Jurusan Teknik Elektro pada Fakultas Sains dan Teknik Universitas Jenderal Soedirman pada tahun 2006.

Semasa kuliah, penulis berkesempatan aktif dalam organisasi mahasiswa diantaranya adalah Badan Eksekutif Mahasiswa (BEM) dan Himpunan Mahasiswa Teknik Elektro (HMTE). Penulis juga salah satu penggagas berdirinya KRONIKA Soedirman (Komunitas Robotik dan Elektronika Soedirman) bersama Aji Setiono S.T, Herryawan Pujiharsono S.T, Irunowo S.T.

Pada jurusan teknik elektro, penulis mengambil konsentrasi kendali dan beberapa kali menjadi asisten praktikum di laboratorium teknik elektro. Penulis juga pernah mengikuti beberapa perlombaan tingkat nasional seperti KRCI (Kontes Robot Cerdas Indonesia), GemasTIK (Pagelaran Mahasiswa Teknik dan Informatika) serta perlombaan robotik yang diselenggarakan beberapa Universitas di Indonesia.

MOTTO

Selalu berusaha menguasai hal - hal yang POSITIF

*A paragraph in the book doesn't give you license to
STOP thinking*

PERSEMBAHAN

Segala puji dan syukur kehadirat Allah SWT atas segala rahmat dan hidayah-Nya, karya kecil ini saya persembahkan untuk:

- Kedua orang tua tercinta yang telah memberikan dukungan dengan segala daya dan upaya hingga saya mampu menyelesaikan studi di Teknik Elektro Universitas Jenderal Soedirman. Bagaimanapun juga baktiku tak akan cukup untuk membalas jasa kalian.
- Kakak ku tercinta yang begitu menantikan aku selesai kuliah. Terimakasih atas bantuannya selama ini.
- Kedua adiku yang manja, maaf selama ini telah menjadi kakak yang kurang baik.
- Kristina Indriyati yang selalu setia mendampingiku dalam menyelesaikan tugas akhir ini. Terima kasih atas support dan kesabarannya, kamu yang selalu membangkitkan semangatku.

UCAPAN TERIMA KASIH

Teriring ketulusan hati, terima kasih kuucapkan untuk :

- Bapak Imron Rosyadi,ST.,MSc dan Azis Wisnu Widhi N,ST.,M.Eng selaku pembimbing I dan II. Terimakasih sebesar – besarnya atas bimbingan bapak.
- Segenap dosen teknik elektro Universitas Jenderal Soedirman yang telah banyak membantu.
- Segenap crew CV. COMUTECH (Ifan, Herry, Bangun, Ipunk dkk) yang selama ini selalu berjuang bersama.
- Teman – teman elektro yang telah banyak membantu.

KATA PENGANTAR

Jaringan Sensor Nirkabel (JSN) semakin berkembang karena sifatnya yang fleksibel dan portabel. Berbagai macam aplikasi dapat dipadukan dengan sistem komunikasi radio untuk membentuk Jaringan Sensor Nirkabel. JSN sangat tepat diaplikasikan di-Indonesia mengingat topografi Indonesia yang berupa kepulauan, daratanya beragam tipe seperti pegunungan, lembah, daratan landai dan sebagainya.

Dalam skripsi ini difokuskan pada aplikasi JSN untuk pengukuran pergerakan material tanah yang mana sangat bermanfaat pada pemantauan daerah rawan bencana longsor. Dengan adanya data pergerakan tanah yang terekam secara kontinyu, maka penanganan terhadap daerah rawan bencana akan lebih mudah. Begitu juga dengan upaya mitigasi bencananya.

Dalam penelitian ini pengukuran pergerakan tanah hanya memanfaatkan akselerometer yang tentunya masih memiliki banyak kelemahan dalam pengukuran. Akan tetapi aplikasi ini dapat memberi gambaran umum dalam perancangan sistem pemantauan bencana tanah longsor yang lebih komplek.

Akuisisi data pada sensor tentunya tak lepas dari adanya derau. Upaya penanganan derau pada akuisisi data sensor secara lengkap dibahas pada bab IV dalam sub bab *Reduksi Error Pada Pengukuran dan Perhitungan*. Pada bab IV juga dibahas sekilas mengenai analisis *Quality Of Service* (QOS) yang mencakup *losses* daya pada komunikasi nirkabel dan konsumsi daya sistem.

Meskipun telah berupaya dengan segenap daya dan upaya, penulis menyadari masih banyak kekurangan dan kekeliruan dalam penyusunan laporan ini sehingga kritik dan saran dari semua pihak sangat diharapkan.

Penulis berharap semoga penulisan laporan ini dapat bermanfaat bagi penulis sendiri serta kalangan civitas akademika dan para pembaca secara keseluruhan.

Purbalingga, Agustus 2013

Penulis

ABSTRAK

Salah satu upaya dalam mitigasi tanah longsor adalah pemantauan pergerakan tanah secara berkala. Hal ini dimungkinkan jika ada perangkat yang memiliki kemampuan untuk mengukur perpindahan tanah. Accelerometer H48C adalah sebuah alat yang mampu mengukur percepatan dinamis dan statis sehingga memungkinkan untuk mengukur perpindahan material tanah. Masalah dalam desain sistem pengukuran perpindahan tanah adalah daerah yang umumnya terletak di lereng bukit atau pegunungan. Pembuatan sistem secara wireline pasti akan memiliki keterbatasan dalam hal fleksibilitas dan portabilitas. Jaringan Sensor Nirkabel (JSN) merupakan salah satu solusi untuk masalah ini. Transfer data pada JSN dapat menggunakan modul RF XBee menggunakan jaringan komunikasi ZigBee.

ABSTRACT

One of the efforts in mitigating landslides are periodic monitoring of soil movement. It is possible if there is a device that have abillity to measuring ground displacement. H48C accelerometer is a device that capable of measuring dynamic and static acceleration so that it is possible to measure the displacement of the soil material. Problem in the design of ground displacement measurement system is a field that is generally located on the slopes of hills or mountains. Manufacture of wireline system will certainly have limitations in terms of flexibility and portability. Wireless Sensor Networks (WSN) is one solution to these problems. Transfer data on WSN can use the XBee RF module using ZigBee communication network.

DAFTAR ISI

LAPORAN TUGAS AKHIR.....	i
LEMBAR PENGESAHAN	ii
TENTANG PENULIS.....	iii
MOTTO	iv
PERSEMPAHAN.....	v
UCAPAN TERIMA KASIH	vi
KATA PENGANTAR.....	vii
ABSTRAK	ix
<i>ABSTRACT</i>	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xx
PENDAHULUAN.....	1
I.1 Latar Belakang	1
I.2 Perumusan Masalah	3
I.3 Batasan Masalah	4
I.4 Tujuan	4
I.5 Manfaat	5
TINJAUAN PUSTAKA	6

II.1	Longsor	6
II.1.1	Jenis - jenis longsor.....	6
II.1.2	Faktor Penyebab Tanah Longsor	10
II.2	Integrasi Numerik	15
II.2.1	Integrasi data diskret	17
II.3	Jaringan Sensor Nirkabel (<i>Wireless Sensor Network</i>)	17
II.3.1	Arsitektur Jaringan Sensor Nirkabel	18
II.3.2	Sensor Percepatan 3 Sumbu.....	20
a.	Prinsip Kerja Akselerometer.....	20
b.	Accelerometer H48C.....	22
c.	Pengukuran Perpindahan Posisi	24
II.4	Sistem Komunikasi Nirkabel (wireless) RF (Radio Frekuensi).....	25
II.4.1	Protokol.....	26
II.4.2	Modul X-BeePro.....	27
II.4.3	Enkripsi / Deskripsi Data	29
II.5	Mikrokontroller.....	31
METODE PENELITIAN	34
III.1	Waktu dan Tempat.....	34
III.2	Alat dan Bahan.....	34
III.3	Metode Penelitian	34
III.1	Studi Pustaka.....	34
III.2	Identifikasi Permasalahan	35
III.3	Perancangan Sistem	35
a.	Desain peralatan.....	35
b.	Perancangan perangkat lunak (software)	41

III.4	Pembuatan Alat.....	44
III.5	Pengujian Sistem.....	45
III.6	Pengaplikasian Sistem.....	46
III.4	Jadwal Penelitian	47
	HASIL DAN PEMBAHASAN	48
IV.1	Unit RTC (Real Time Clock).....	48
IV.1.1	Skematik Rangkaian RTC.....	48
IV.1.2	Algoritma Penggunaan RTC (<i>Real Time Clock</i>) dengan IC DS1307	49
IV.2	Metode Akusisi Data Percepatan	55
IV.2.1	Skematik Rangkaian <i>Accelerometer</i>	57
IV.2.2	Algoritma Pembacaan Percepatan Tiap Sumbu	58
IV.2.3	Karakteristik Akselerometer H48C	68
a.	Percepatan Statik.....	68
b.	Pengujian Percepatan Statik.....	69
c.	Percepatan Dinamik	76
d.	Pengujian Percepatan Dinamik	78
e.	Analisa Hasil Pengujian Percepatan Dinamik.....	90
IV.2.4	Reduksi Error Pada Pengukuran Dan Perhitungan	97
a.	Penentuan Orde Filter	98
b.	Penerapan Pada Algoritma Program	109
IV.3	Unit Komunikasi (modul RF)	113
IV.3.1	Skematik rangkaian modul RF pada router.....	114
IV.3.2	Algoritma Pengiriman Data Pada Router.....	116
IV.3.3	Skematik rangkaian modul RF pada gateway	123
IV.3.4	Algoritma pembacaan data pada gateway.....	126

IV.4	Pengujian RSSI (<i>Receive Signal Strength Indicator</i>).....	135
IV.5	Konsumsi Daya	143
PENUTUP.....		146
IV.1	Kesimpulan	146
IV.2	Saran	146
DAFTAR PUSTAKA		147
LAMPIRAN - LAMPIRAN		149

DAFTAR GAMBAR

<i>Gambar 2.1 Longsor translasi</i>	7
<i>Gambar 2.2 Longsor rotas</i>	7
<i>Gambar 2.3 Pergerakan blok</i>	8
<i>Gambar 2.4 Pergerakan batu</i>	8
<i>Gambar 2.5 Rayapan tanah</i>	9
<i>Gambar 2.6 Aliran bahan rombakan</i>	9
<i>Gambar 2.7 Arsitekture individual node</i>	18
<i>Gambar 2.8 Konfigurasi sebuah jaringan sensor sederhana</i>	19
<i>Gambar 2.9 Topologi star pada jaringan sensor nirkabel</i>	19
<i>Gambar 2.10 Topologi mesh pada jaringan sensor nirkabel</i>	20
<i>Gambar 2.11 Topologi hybrid star – mesh</i>	20
<i>Gambar 2.12 Perinsip kerja akselerometer</i>	21
<i>Gambar 2.13 Analogi pergerakan sensor akselerometer 3 sumbu</i>	22
<i>Gambar 2.14 Skematik konfigurasi sederhana dari H48C.....</i>	23
<i>Gambar 2.15 Diagram blok ZeroG detector.....</i>	24
<i>Gambar 2.16 Aturan penulisan AT Command</i>	29
<i>Gambar 2.17 Ilustrasi metode Caesar Chipper</i>	30
<i>Gambar 2.18 Block diagram arsitektur dasar mikrokontroler</i>	32
<i>Gambar 2.19 Arsitektur mikrokontroler Atmega8</i>	33

<i>Gambar 3.1 Diagram blok unit elektronik (kiri:gateway, kanan:router).....</i>	36
<i>Gambar 3.2 Pengkondisian sinyal pada akselerometer H48C</i>	36
<i>Gambar 3.3 Skematik arduino deumilanove</i>	37
<i>Gambar 3.4 Konfigurasi pin pada modul Xbee dengan arduino board</i>	39
<i>Gambar 3.5 Skematik rangkaian RTC menggunakan DS1307</i>	40
<i>Gambar 3.6 Skematik regulator tegangan menggunakan IC 7805</i>	40
<i>Gambar 3.7 Flowchart program pada router</i>	42
<i>Gambar 3.8 Flowchart program pada gateway</i>	43
<i>Gambar 3.9 Flowchart program pada komputer server</i>	44
<i>Gambar 3.10 Diagram alir metode penelitian</i>	47
<i>Gambar 4.1 Skematik rangkaian modul RTC dengan IC DS1307.....</i>	49
<i>Gambar 4.2 Pembacaan data DS1307</i>	55
<i>Gambar 4.3 Blok diagram H48C</i>	56
<i>Gambar 4.4 Diagram blok MCP3204</i>	57
<i>Gambar 4.5 Skematik rangkaian pengujian akselerometer</i>	58
<i>Gambar 4.6 Timing diagram MCP3204</i>	59
<i>Gambar 4.7 Hasil pembacaan modul akselerometer</i>	66
<i>Gambar 4.8 Ilustrasi pengambilan data sebanyak 6 sampel untuk tiap sumbu.....</i>	67
<i>Gambar 4.9 Sudut roll (ϕ) dalam plot dua dimensi antara sumbu Y dan Z (gambar a), sudut pitch(ρ) dalam dua dimensi antara sumbu X dan Z (gambar b).....</i>	69

<i>Gambar 4.10 Grafik percepatan pada akselerometer bergerak statis dengan $Ay \approx 0$ bergerak 3600</i>	71
<i>Gambar 4.11 Grafik percepatan pada akselerometer bergerak statis dengan $Ay \approx 0$ bergerak 900 bolak – balik</i>	71
<i>Gambar 4.12 Grafik percepatan pada akselerometer bergerak statis dengan $Ax \approx 0$ bergerak 3600</i>	72
<i>Gambar 4.13 Grafik percepatan pada akselerometer bergerak statis dengan $Ax \approx 0$ bergerak 900 bolak – balik</i>	72
<i>Gambar 4.14 Nilai yang didapat apabila menggunakan persamaan 4.8 untuk menghitung sudut ρ dari data akselerometer pada tabel 4.2</i>	73
<i>Gambar 4.15 Penentuan daerah kuadran dengan menganalisa data vektor percepatan sumbu X dan Y terhadap sumbu Z</i>	75
<i>Gambar 4.16 Nilai sudut ρ yang dihasilkan apabila posisi kuadran diketahui</i>	76
<i>Gambar 4.17 Posisi modul pada saat pengujian pergerakan tiap sumbu</i>	81
<i>Gambar 4.18 Grafik pengujian pergerakanmodul pada sumbu x positif</i>	83
<i>Gambar 4.19 Grafik pengujian pergerakanmodul pada sumbu x negatif</i>	84
<i>Gambar 4.20 Grafik pengujian pergerakanmodul pada sumbu y positif</i>	85
<i>Gambar 4.21 Grafik pengujian pergerakanmodul pada sumbu y negatif</i>	86
<i>Gambar 4.22 Grafik pengujian pergerakanmodul pada sumbu z positif</i>	88
<i>Gambar 4.23 Grafik pengujian pergerakanmodul pada sumbu z negatif</i>	90
<i>Gambar 4.24 Perbedaan arah pergerakan benda pada arah positif (kiri) dan arah negatif(kanan)</i>	90
<i>Gambar 4.25 Hasil integrasi data percepatan pada pergerakan benda pada arah positif (kiri) dan arah negatif(kanan)</i>	91

<i>Gambar 4.26 Hasil integrasi data kecepatan pada arah positif (kiri) dan arah negatif (kanan)</i>	92
<i>Gambar 4.27 Perbandingan hasil pengujian dengan data ideal</i>	94
<i>Gambar 4.28 Analogi Micro Electro-Mechanical System yang diadopsi oleh akselerometer H48C</i>	95
<i>Gambar 4.29 Grafik nilai percepatan sebelum dan sesudah menggunakan filter orde 1 hingga 10</i>	106
<i>Gambar 4.30 Grafik nilai rMSE hasil integrasi kedua (data perpindahan) pada tiap – tiap orde filter dan data yang tidak menggunakan filter.....</i>	107
<i>Gambar 4.31 Grafik cross axis effect pada pengujian tidak menggunakan filter dan menggunakan filter orde 1 hingga 10</i>	108
<i>Gambar 4.32 Hasil pengujian akselerometer setelah difilter</i>	112
<i>Gambar 4.33 Skematik perantaraan Xbee Pro Series 1 dengan arduino deumilanove pada router</i>	114
<i>Gambar 4.34 Xbee Shield yang telah terpasang Xbee Pro Series 1</i>	115
<i>Gambar 4.35 Flowchart algoritma pewaktuan pengiriman data (a) menggunakan counter internal (b) menggunakan RTC</i>	121
<i>Gambar 4.36 Skematik rangkaian pada gateway</i>	125
<i>Gambar 4.37 Jumper RX dan TX pada blok APC dari arduino IO Expansion dengan pin 12 dan pin 13</i>	126
<i>Gambar 4.38 Tipe frame API untuk Xbee RX packet pada pengalamanan 16 bit(Maxstream.Inc,2007)</i>	127
<i>Gambar 4.39 Data yang diterima oleh modul Xbee</i>	129
<i>Gambar 4.40 Hasil runing program pada gateway</i>	135

<i>Gambar 4.41 Skematik rangkaian pengujian RSSI</i>	136
<i>Gambar 4.42 Flowchart algoritma modul penerima (kiri) dan modul pengirim (kanan) yang digunakan pada pengujian RSSI</i>	137
<i>Gambar 4.43 Tinggi modul dari permukaan tanah (kiri :pengirim, kanan:penerima).....</i>	138
<i>Gambar 4.44 Grafik perbandingan nilai path loss terhadap jarak</i>	140
<i>Gambar 4.45 Lokasi pengujian RSSI pada medan interferensi.....</i>	141
<i>Gambar 4.46 Grafik perbandingan nilai path loss terhadap jarak pada pengujian modul ter-interferensi</i>	142
<i>Gambar 4.47 Skematik pengukuran daya dari sebuah router</i>	143

DAFTAR TABEL

Tabel 3.1 Jadwal penelitian.....	38
Tabel 4.1 Konfigurasi bit kontrol MCP324.....	60
Tabel 4.2 Data pengujian pergerakan sumbu x arah positif.....	82
Tabel 4.3 Data pengujian pergerakan sumbu x arah negatif.....	83
Tabel 4.4 Data pengujian pergerakan sumbu y arah positif.....	84
Tabel 4.5 Data pengujian pergerakan sumbu y arah negatif.....	85
Tabel 4.6 Data pengujian pergerakan sumbu z arah positif.....	87
Tabel 4.7 Data pengujian pergerakan sumbu z arah negatif.....	88
Tabel 4.8 Data hasil pengujian pergerakan akselerometer pada sumbu x, y, dan sumbu z.....	96
Tabel 4.9 Hasil double integrasi data percepatan pada sumbu yang tidak bergerak.....	96
Tabel 4.10 Koefisien rekursi filter besel orde 1 hingga 10 dengan frekuensi cut off 5 Hz dan frekuensi sampling 125 Hz	99
Tabel 4.11 Nilai rMSE pada pengujian filter dari orde 1 hingga 10 dan data yang tidak difilter	100
Tabel 4.12 Cross axis effect pada pengujian tidak menggunakan filter dan menggunakan filter orde 1 hingga 10.....	108
Tabel 4.13 Data hasil pengujian pergerakan akselerometer pada sumbu x, y, dan z setelah difilter.....	112

Tabel 4.14 Hasil double integrasi data percepatan pada sumbu yang tidak bergerak setelah difilter.....	113
Tabel 4.15 Tabel setting parameter Xbee untuk pengujian RSSI.....	136
Tabel 4.16 Data hasil pengujian RSSI modul terhubung secara LOS.....	138
Tabel 4.17 Data hasil pengujian RSSI pada medan berpenghalang.....	141
Tabel 4.18 Hasil pengukuran konsumsi arus pada router.....	144

BAB I

PENDAHULUAN

I.1 Latar Belakang

Berdasarkan tinjauan aspek geologis, jenis tanah yang sering dijumpai di Indonesia adalah hasil letusan gunung api. Tanah ini memiliki komposisi sebagian besar lempung dengan sedikit pasir dan bersifat subur. Tanah pelapukan yang berada di atas batuan kedap air pada pertumbuhan / punggungan dengan kemiringan sedang hingga terjal berpotensi mengakibatkan tanah longsor pada musim hujan dengan curah hujan berkualitas tinggi. Jika pertumbuhan tersebut tidak ada tanaman keras berakar kuat dan dalam, maka kawasan tersebut rawan bencana tanah longsor.

Tercatat selama tahun 2011 terjadi 210 kejadian tanah longsor dengan yang mengakibatkan 177 korban meninggal, 837 orang menderita dan mengungsi dan kerusakan bangunan mencapai 2992 rumah dan bangunan (BNPB, 2012^a). Angka kejadian bencana ini terbilang sangat tinggi sehingga diperlukan penanganan khusus terhadap upaya penanggulangan dan mitigasi bencana terhadap tanah longsor.

Salah satu parameter dari tanah longsor adalah kecepatan pergerakan tanah (BNPB, 2011). Yang mana kecepatan merupakan besaran turunan dari jarak perpindahan. Salah satu metode yang dapat dilakukan untuk mendeteksi adanya perpindahan material tanah adalah dengan cara melakukan pengukuran perpindahan posisi tanah secara berkala.

Pada saat ini didalam sistem instrumentasi elektronik telah dikembangkan sensor percepatan yang mana percepatan merupakan turunan dari kecepatan pergerakan sehingga dengan menggunakan teknik integrasi hasil pembacaan sensor percepatan dapat diperoleh nilai dari kecepatan. Sebagai contoh modul *accelerometer* H48C buatan Parallax mampu membaca percepatan pada 3 sumbu axial. Sehingga besar kemungkinan sensor percepatan H48C dapat diaplikasikan untuk pengukuran perpindahan material tanah secara berkala.

Dalam areal material longsor, jarak perpindahan material dari suatu daerah dengan daerah yang lain beraneka ragam dan umumnya daerah yang rawan bencana longsor merupakan daerah perbukitan sehingga perlu adanya suatu perangkat yang mampu melakukan pengukuran terhadap jarak perpindahan material tersebut pada titik yang berbeda secara bersamaan. Titik – titik yang dianggap perlu dilakukan pengukuran umumnya berjarak relatif jauh antara yang satu dengan lainnya, sehingga penggunaan kabel untuk pertukaran data antar perangkat mempunyai beberapa keterbatasan.

Robert Faludi (2010) dari New York University, USA, mengaplikasikan jaringan Zigbee¹ sebagai jaringan sensor nirkabel dengan memanfaatkan modul XBee sebagai modul komunikasinya. Yang mana dalam jaringan sensor nirkabel ini komunikasi data tiap – tiap *mote*² dilakukan dengan menggunakan frekuensi radio sehingga memungkinkan dilakukan pertukaran data antar *mote* pada jarak yang relatif jauh tanpa menggunakan kabel. Sehingga

¹ Standar protokol jaringan wireless yang ditujukan untuk otomasi dan kontrol jarak jauh dengan konsumsi daya dan *transfer rate* yang relatif rendah.

² Unit dari jaringan sensor nirkabel yang tersusun dari modul komunikasi, kontroler, memori, sensor – sensor, dan catu daya.

jaringan Zigbee memungkinkan untuk digunakan sebagai jaringan sensor nirkabel untuk pembacaan kecepatan pergerakan tanah.

Robert Faludi juga menegaskan bahwa, modul wireless Xbee yang digunakan mampu menangani input dan output secara langsung karena modul telah terintegrasi dengan mikrokontroler. Akan tetapi penggunaan modul xbee secara *standalone* dalam sebuah *mote* memiliki beberapa keterbatasan diantaranya adalah kurangnya fleksibilitas dalam pemrograman, keterbatasan fungsi input – output modul, proses kalkulasi pada mikrokontroler internal dapat mengganggu stabilitas frekuensi pengiriman maupun penerimaan data sehingga untuk menanggulanginya dibutuhkan mikrokontroler eksternal. Beberapa modul mikrokontroler yang cukup terkenal dan banyak digunakan dalam aplikasi sistem embeded adalah Arduino, PIC Chip, BASICstamp, Beagle Board.

I.2 Perumusan Masalah

Dari uraian di atas maka dapat dirumuskan permasalahan sebagai berikut.

1. Bagaimana memanfaatkan sensor percepatan untuk akuisisi data kecepatan pergerakan tanah?
2. Bagaimana melakukan kalibrasi dan menentukan tingkat akurasi sensor percepatan dalam melakukan pengukuran pergerakan tanah?
3. Bagaimana memanfaatkan jaringan sensor nirkabel menggunakan protokol Zigbee untuk komunikasi data percepatan pergerakan tanah antar *mote*?

4. Bagaimana sinkronisasi data antara akselerometer, mikrokontroler eksternal, dan modul wireless?

I.3 Batasan Masalah

Dalam pembuatan skripsi ini penulis hanya akan membahas sebagai berikut:

1. Perangkat yang digunakan adalah sebagai berikut.
 - a. Sensor percepatan menggunakan modul *accelerometer H48C*.
 - b. Mikrokontroler eksternal menggunakan Arduino type Arduino Deumilanove.
 - c. Modul wireless menggunakan Xbee Pro *Series 1*.
2. Bagaimana merancang sebuah sistem akusisi data yang mampu melakukan pengambilan data dari lingkungan sekaligus berkomunikasi dengan perangkat lain.
3. Topologi jaringan ZigBee yang digunakan adalah topologi *star* dan tidak membahas detil tentang topologi jaringan.
4. Perancangan system tidak membahas masalah keamanan jaringan.

I.4 Tujuan

Tujuan dilakukannya penelitian ini yaitu :

1. Melakukan pengukuran terhadap pergerakan material longsor menggunakan sensor percepatan.

2. Menemukan tingkat akurasi terhadap modul sensor akselerometer H48C.
3. Mendapatkan parameter pengkondisian sinyal yang sesuai untuk akselerometer H48C.
4. Merancang dan membuat sistem *telemetry* berbasis *Zigbee wireless RF*.
5. Mengetahui konsumsi daya sebuah sistem jaringan sensor nirkabel *Zigbee*.

I.5 Manfaat

Manfaat dari melakukan penelitian ini yaitu :

1. Mengaplikasikan jaringan sensor secara nirkabel terhadap sistem peringatan dini bencana tanah longsor.
2. Untuk menyelesaikan permasalahan dari sistem peringatan dini bencana tanah longsor yang ada sebelumnya yang masih terkendala masalah fleksibilitas dan portabilitas.
3. Dapat melakukan alih teknologi tentang jaringan sensor nirkabel yang masih terkategori baru untuk Indonesia.
4. Sebagai bahan kajian untuk aplikasi jaringan sensor nirkabel dibidang lain.

BAB II

TINJAUAN PUSTAKA

II.1 Longsor

Tanah longsor adalah perpindahan material pembentuk lereng berupa batuan, bahan rombakan, tanah, atau material campuran tersebut, bergerak ke bawah atau keluar lereng. Proses terjadinya tanah longsor dapat diterangkan sebagai berikut: air yang meresap ke dalam tanah akan menambah bobot tanah. Jika air tersebut menembus sampai tanah kedap air yang berperan sebagai bidang gelincir, maka tanah menjadi licin dan tanah pelapukan di atasnya akan bergerak mengikuti lereng dan keluar lereng.

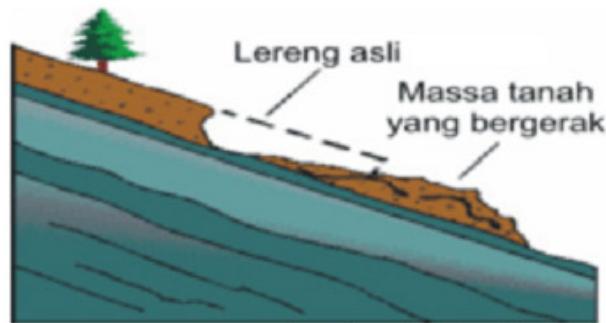
II.1.1 Jenis - jenis longsor

Ada 6 jenis tanah longsor (BNPB, 2011), yakni: longsoran translasi, longsoran rotasi, pergerakan blok, runtuhannya batu, rayapan tanah, dan aliran bahan rombakan. Jenis longsoran translasi dan rotasi paling banyak terjadi di Indonesia. Sedangkan longsoran yang paling banyak memakan korban jiwa manusia adalah aliran bahan rombakan.

1. Longsoran Translasi.

Longsoran translasi adalah bergeraknya massa tanah dan batuan pada bidang gelincir berbentuk rata atau menggelombang landai.

Ilustrasi dari longsor translasi dapat dilihat pada gambar 2.1.



Gambar 2.1 Longsor translasi

2. Longsoran Rotasi.



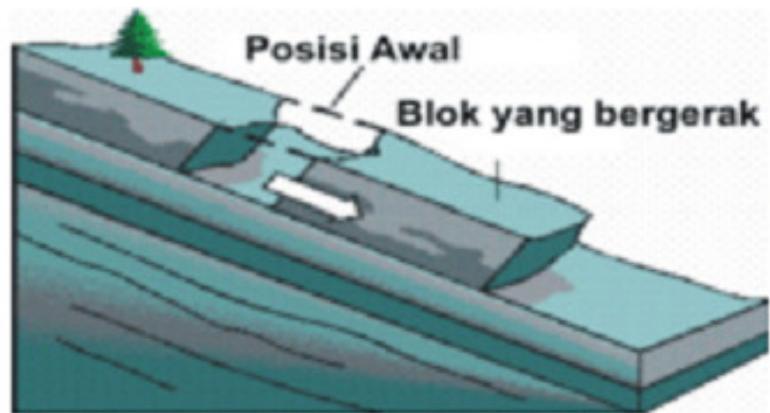
Gambar 2.2 Longsor rotasi

Longsoran rotasi adalah bergeraknya massa tanah dan batuan pada bidang gelincir berbentuk cekung. Ilustrasi dari longsor rotasi dapat dilihat pada gambar 2.2.

3. Pergerakan Blok

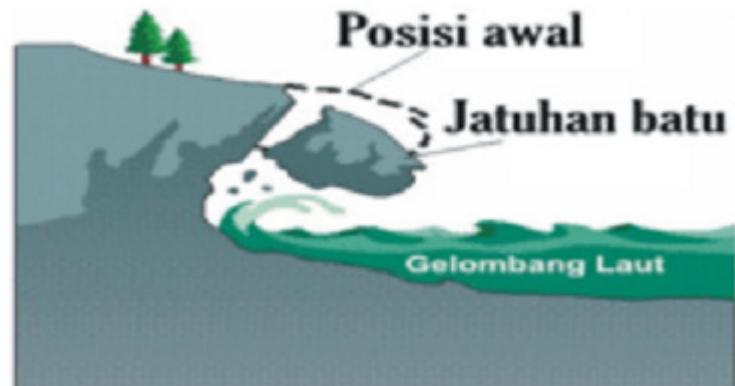
Pergerakan blok adalah perpindahan batuan yang bergerak pada bidang gelincir berbentuk rata. Longsoran ini disebut juga longsoran translasi blok batu.

Ilustrasi dari pergerakan blok dapat dilihat pada gambar 2.3.



Gambar 2.3 Pergerakan blok

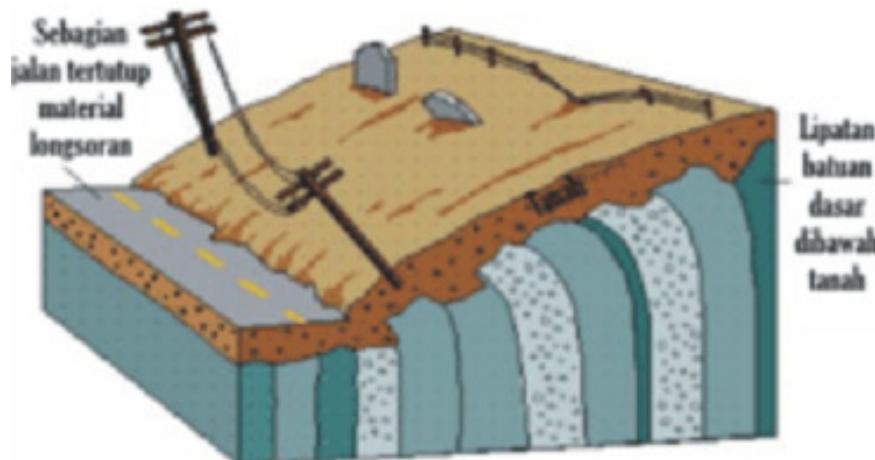
4. Runtuhan Batu



Gambar 2.4 Pergerakan batu

Runtuhan batu terjadi ketika sejumlah besar batuan atau material lain bergerak ke bawah dengan cara jatuh bebas. Umumnya terjadi pada lereng yang terjal hingga menggantung terutama di daerah pantai. Batu-batu besar yang jatuh dapat menyebabkan kerusakan yang parah. Ilustrasi dari pergerakan batu dapat dilihat pada gambar 2.4.

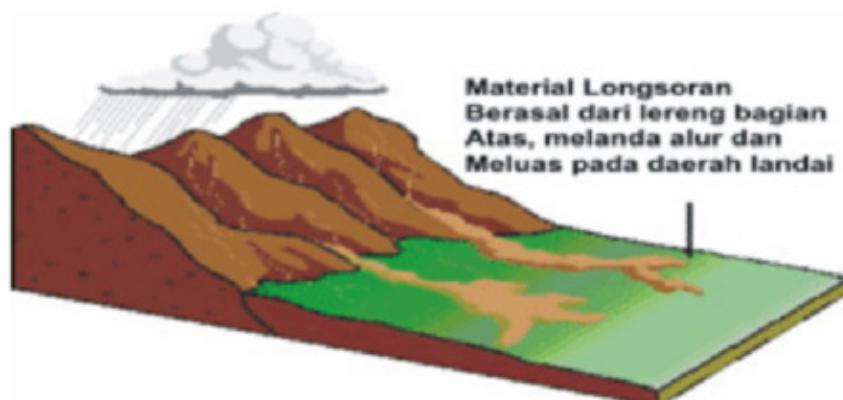
5. Rayapan Tanah



Gambar 2.5 Rayapan tanah

Rayapan Tanah adalah jenis tanah longsor yang bergerak lambat. Jenis tanahnya berupa butiran kasar dan halus. Jenis tanah longsor ini hampir tidak dapat dikenali. Setelah waktu yang cukup lama longsor jenis rayapan ini bisa menyebabkan tiang-tiang telepon, pohon, atau rumah miring ke bawah. Ilustrasi dari rayapan tanah dapat dilihat pada gambar 2.5.

6. Aliran Bahan Rombakan



Gambar 2.6 Aliran bahan rombakan

Jenis tanah longsor ini terjadi ketika massa tanah bergerak didorong oleh air. Kecepatan aliran tergantung pada

kemiringan lereng, volume dan tekanan air, dan jenis materialnya. Gerakannya terjadi di sepanjang lembah dan mampu mencapai ratusan meter jauhnya. Di beberapa tempat bisa sampai ribuan meter seperti di daerah aliran sungai di sekitar gunungapi. Aliran tanah ini dapat menelan korban cukup banyak. Ilustrasi dari aliran bahan rombak dapat dilihat pada gambar 2.6.

II.1.2 Faktor Penyebab Tanah Longsor

Pada prinsipnya tanah longsor terjadi bila gaya pendorong pada lereng lebih besar daripada gaya penahan. Gaya penahan umumnya dipengaruhi oleh kekuatan batuan dan kepadatan tanah. Sedangkan gaya pendorong dipengaruhi oleh besarnya sudut lereng, air, beban serta berat jenis tanah batuan.

Faktor-faktor Penyebab Tanah Longsor :

1. Hujan

Ancaman tanah longsor biasanya dimulai pada bulan November karena meningkatnya intensitas curah hujan. Musim kering yang panjang akan menyebabkan terjadinya penguapan air di permukaan tanah dalam jumlah besar. Hal itu mengakibatkan munculnya pori-pori atau rongga tanah hingga terjadi retakan dan merekahnya tanah permukaan.

Ketika hujan, air akan menyusup ke bagian yang retak sehingga tanah dengan cepat mengembang kembali. Pada awal

musim hujan, intensitas hujan yang tinggi biasanya sering terjadi, sehingga kandungan air pada tanah menjadi jenuh dalam waktu singkat.

Hujan lebat pada awal musim dapat menimbulkan longsor, karena melalui tanah yang merekah air akan masuk dan terakumulasi di bagian dasar lereng, sehingga menimbulkan gerakan lateral. Bila ada pepohonan di permukaannya, tanah longsor dapat dicegah karena air akan diserap oleh tumbuhan.

Akar tumbuhan juga akan berfungsi mengikat tanah.

2. Lereng terjal

Lereng atau tebing yang terjal akan memperbesar gaya pendorong. Lereng yang terjal terbentuk karena pengikisan air sungai, mata air, air laut, dan angin. Kebanyakan sudut lereng yang menyebabkan longsor adalah 180° apabila ujung lerengnya terjal dan bidang longsorannya mendatar.

3. Tanah yang kurang padat dan tebal

Jenis tanah yang kurang padat adalah tanah lempung atau tanah liat dengan ketebalan lebih dari 2,5 m dan sudut lereng lebih dari 220°. Tanah jenis ini memiliki potensi untuk terjadinya tanah longsor terutama bila terjadi hujan. Selain itu tanah ini sangat rentan terhadap pergerakan tanah karena menjadi lembek terkena air dan pecah ketika hawa terlalu panas.

4. Batuan yang kurang kuat

Batuan endapan gunung api dan batuan sedimen berukuran pasir dan campuran antara kerikil, pasir, dan lempung umumnya kurang kuat. Batuan tersebut akan mudah menjadi tanah bila mengalami proses pelapukan dan umumnya rentan terhadap tanah longsor bila terdapat pada lereng yang terjal.

5. Jenis tata lahan

Tanah longsor banyak terjadi di daerah tata lahan persawahan, perladangan, dan adanya genangan air di lereng yang terjal. Pada lahan persawahan akarnya kurang kuat untuk mengikat butir tanah dan membuat tanah menjadi lembek dan jenuh dengan air sehingga mudah terjadi longsor. Sedangkan untuk daerah perladangan penyebabnya adalah karena akar pohonnya tidak dapat menembus bidang longsoran yang dalam dan umumnya terjadi di daerah longsoran lama.

6. Getaran

Getaran yang terjadi biasanya diakibatkan oleh gempabumi, ledakan, getaran mesin, dan getaran lalulintas kendaraan. Akibat yang ditimbulkannya adalah tanah, badan jalan, lantai, dan dinding rumah menjadi retak.

7. Susut muka air danau atau bendungan

Akibat susutnya muka air yang cepat di danau maka gaya penahan lereng menjadi hilang, dengan sudut kemiringan

waduk 220 mudah terjadi longsoran dan penurunan tanah yang biasanya diikuti oleh retakan.

8. Adanya beban tambahan

Adanya beban tambahan seperti beban bangunan pada lereng, dan kendaraan akan memperbesar gaya pendorong terjadinya longsor, terutama di sekitar tikungan jalan pada daerah lembah. Akibatnya adalah sering terjadinya penurunan tanah dan retakan yang arahnya ke arah lembah.

9. Pengikisan/erosi

Pengikisan banyak dilakukan oleh air sungai ke arah tebing. Selain itu akibat penggundulan hutan di sekitar tikungan sungai, tebing akan menjadi terjal.

10. Adanya material timbunan pada tebing

Untuk mengembangkan dan memperluas lahan pemukiman umumnya dilakukan pemotongan tebing dan penimbunan lembah. Tanah timbunan pada lembah tersebut belum terpadatkan sempurna seperti tanah asli yang berada di bawahnya.

Sehingga apabila hujan akan terjadi penurunan tanah yang kemudian diikuti dengan retakan tanah.

11. Bekas longsoran lama

Longsoran lama umumnya terjadi selama dan setelah terjadi pengendapan material gunung api pada lereng yang relatif

terjal atau pada saat atau sesudah terjadi patahan kulit bumi.

Bekas longsoran

lama memiliki ciri:

- a. Adanya tebing terjal yang panjang melengkung membentuk tapal kuda. Umumnya dijumpai mata air, pepohonan yang relatif tebal karena tanahnya gembur dan subur.
- b. Daerah badan longsor bagian atas umumnya relatif landai.
- c. Dijumpai longsoran kecil terutama pada tebing lembah.
- d. Dijumpai tebing-tebing relatif terjal yang merupakan bekas longsoran kecil pada longsoran lama.
- e. Dijumpai alur lembah dan pada tebingnya dijumpai retakan dan longsoran kecil.

Longsoran lama ini cukup luas.

12. Adanya bidang diskontinuitas (bidang tidak sinambung)

Bidang tidak sinambung ini memiliki ciri:

- a. Bidang perlapisan batuan
- b. Bidang kontak antara tanah penutup dengan batuan dasar
- c. Bidang kontak antara batuan yang retak-retak dengan batuan yang kuat.

- d. Bidang kontak antara batuan yang dapat melewatkannya air dengan batuan yang tidak melewatkannya air (kedap air).
- e. Bidang kontak antara tanah yang lembek dengan tanah yang padat.
- f. Bidang-bidang tersebut merupakan bidang lemah dan dapat berfungsi sebagai bidang luncuran tanah longsor.

13. Penggundulan hutan

Tanah longsor umumnya banyak terjadi di daerah yang relatif gundul dimana pengikatan air tanah sangat kurang.

14. Daerah pembuangan sampah

Penggunaan lapisan tanah yang rendah untuk pembuangan sampah dalam jumlah banyak dapat mengakibatkan tanah longsor apalagi ditambah dengan guyuran hujan, seperti yang terjadi di Tempat Pembuangan Akhir (TPA) Sampah Leuwigajah di Cimahi. Bencana ini menyebabkan sekitar 120 orang lebih meninggal.

II.2 Integrasi Numerik

Integrasi numerik umumnya dilakukan apabila :

- 1 Fungsi yang akan diintegrasikan sedemikian hingga tidak ada metode analitik untuk menyelesaiakannya.

2 Fungsi yang akan diintegrasi, bentuk eksplisitnya tak diketahui, tetapi diberikan nilai-nilai variabel bebasnya dan nilai-nilai fungsi yang berkorespondensi di dalam suatu interval $[a...b]$.

Masalah umum dari integrasi numerik dapat dinyatakan sebagai berikut:

Diberikan sekumpulan titik $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ dari fungsi $y=f(x)$, dimana bentuk eksplisit dari $f(x)$ tidak diketahui, dan dari data (keterangan) tersebut akan dihitung nilai integral tentu berikut:

seperti didalam diferensiasi numerik, $f(x)$ akan diaproksimasi oleh interpolasi polinom $\theta(x)$, dan hasilnya pada integrasi tersebut adalah nilai aproksimasi integral tentu. Jadi, perbedaan formula integrasi bergantung pada bentuk dari *selisih maju dari Newton*.

Misalkan interval $[a...b]$ dibagi menjadi n interval bagian, sedemikian hingga $a = x_1 < x_2 < x_3 < \dots < x_n = b$. Oleh karena itu $x_n = x_0 + nh$. Dengan demikian diperoleh.

$$I = \int_{x_0}^{x_n} y \, dx \dots \quad (2.2)$$

Aproksimasi y oleh formula selisih maju Newton, kita peroleh:

$$I = \int_{x_0}^{x_n} \left[y_0 + \Delta P y_0 + \frac{p(p-1)}{2} \Delta^2 y_0 + \frac{p(p-1)(p-2)}{6} \Delta^3 y_0 + \dots \right] dx \dots \quad (2.2)$$

Karena $x = x.ph$ maka $dx = h \, dp$ jika disubtitusikan ke persamaan 2.2 maka akan menghasilkan persamaan 2.3 berikut.

$$\int_{x_0}^{x_n} y \, dx = \left[y_0 + \Delta P y_0 + \frac{n}{2} \Delta^2 y_0 + \frac{n(2n-3)}{12} \Delta^3 y_0 + \frac{n(n-2)^2}{24} \Delta^3 y_0 + \dots \right] \dots \dots \dots \quad (2.3)$$

II.2.1 Integrasi data diskret

Data diskret adalah sekumpulan data yang diambil dengan interval pengambilan sampel tertentu. Pada proses integrasi diskrit, perhitungan jauh lebih sederhana karena proses integrasi hanyalah proses penjumlahan. Jika $y_{(n)}$ adalah sekumpulan data diskrit dengan n [a...b], sedangkan selang waktu kejadian antara data ke- n dengan data ke- $(n-1)$ adalah Δt , maka integral dari $y_{(n)}$ adalah $x_{(n)}$ sesuai dengan persamaan 2.4.

II.3 Jaringan Sensor Nirkabel (*Wireless Sensor Network*)

Jaringan Sensor Nirkabel (*Wireless Sensor Network*) merupakan suatu jaringan nirkabel yang terdiri dari beberapa sensor (*node sensor*) yang diletakan diberbagai lokasi yang berbeda untuk memonitoring suatu plan.

Konsep dasar perancangan Jaringan Sensor Nirkabel yaitu memadukan fungsi sensing dari suatu piranti dengan CPU (*Central Processing Unit*) sebagai unit pengolahan dan perhitungan data kemudian dipadukan dengan RF (*Radio Frequency*) untuk komunikasi data.

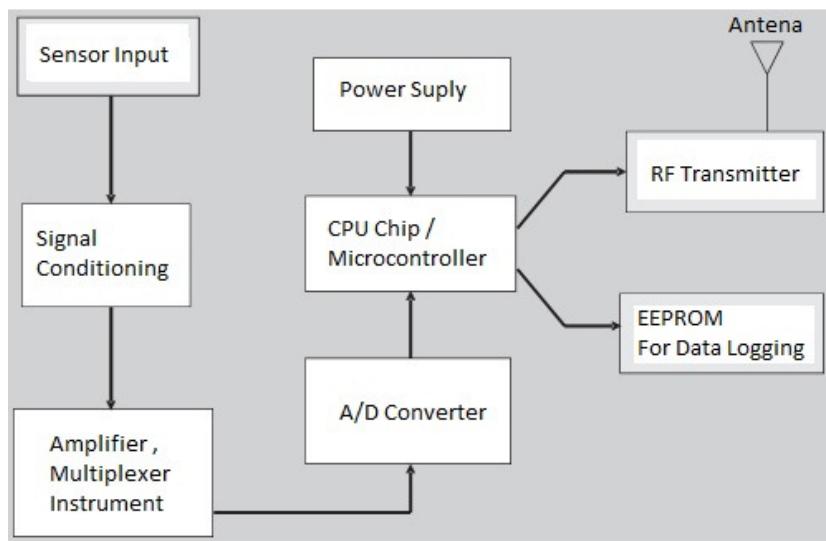
Sensor yang digunakan beraneka ragam sesuai dengan kebutuhan. Sensor-sensor tersebut akan melakukan akusisi data objek dan mengirimnya ke *gateway* kemudian ke server.

II.3.1 Arsitektur Jaringan Sensor Nirkabel

Komponen utama jaringan sensor nirkabel yaitu :

1. Node / Mote

Node merupakan komponen yang berfungsi untuk pembacaan data lingkungan, penyimpanan data , serta pengiriman data. Sehingga sebuah *node* minimal harus didukung oleh fungsi *sensing*, *data logging* dan *data transmit*. Arsitektur sebuah *node* dapat dilihat dari gambar 2.7 berikut.



Gambar 2.7 Arsitekture individual node

2. Gateway / Base Station

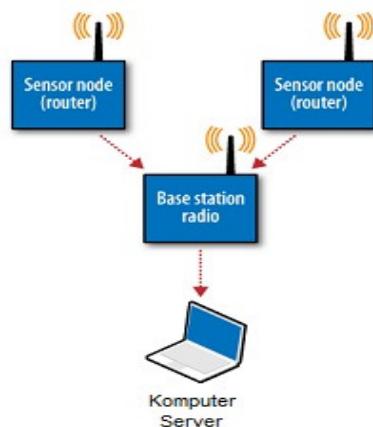
Base station memiliki fungsi utama sebagai pengumpul data dari *node – node* yang tersebar di lapangan kemudian mengirimkanya ke komputer server.

3. Komputer Server

Merupakan piranti yang berfungsi sebagai pusat basis data, sistem jaringan dan penyedia aplikasi. Dengan adanya

komputer server memungkinkan data yang dikirimkan oleh *gateway* terhubung dengan internet, dilakukan pengolahan data untuk keperluan prediksi dan analisis, serta monitoring *node* yang tersebar dilapangan.

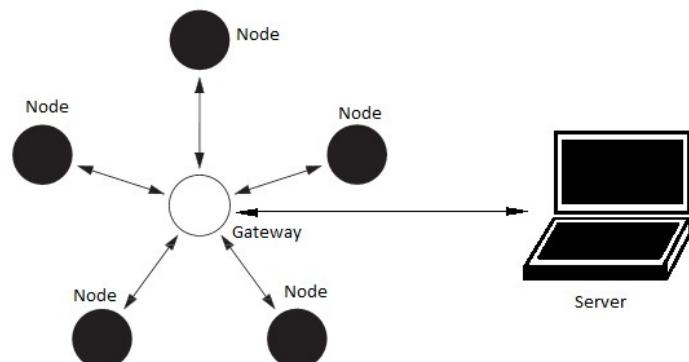
Desain minimum dari sebuah jaringan sensor nirkabel dapat dilihat dari gambar 2.8 berikut :



Gambar 2.8 Konfigurasi Sebuah Jaringan Sensor Nirkabel Sederhana

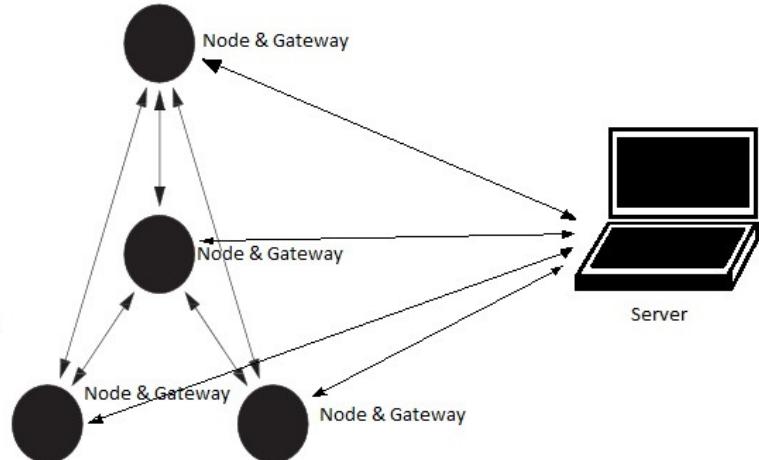
Konfigurasi ketiga komponen tersebut (*node*, *gateway*, dan *server*) dapat beraneka ragam sesuai dengan kebutuhan. Diantaranya adalah sebagai berikut .

1. Topologi Star / Bintang (Gambar 2.9)



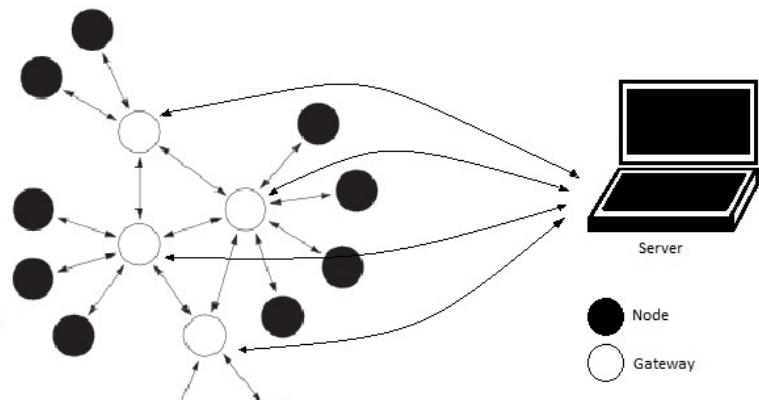
Gambar 2.9 Topologi Star pada Jaringan Sensor Nirkabel

2. Topologi Mesh (Gambar 2.10)



Gambar 2.10 Topologi Mesh pada Jaringan Sensor Nirkabel

3. Hybrid Star - Mesh (Gambar 2.11)



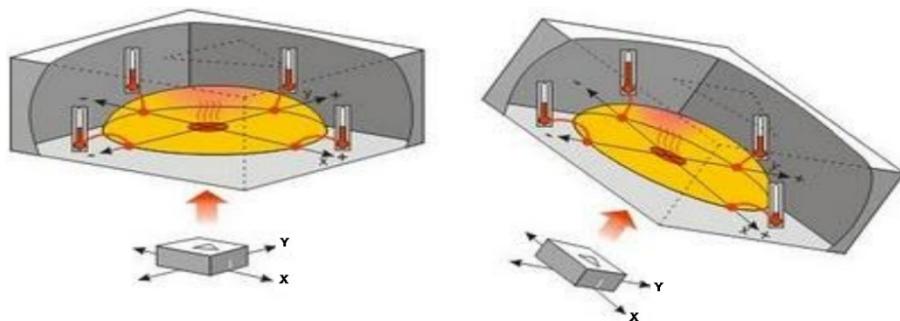
Gambar 2.11 Topologi Hybrid Star – Mesh

II.3.2 Sensor Percepatan 3 Sumbu

a. Prinsip Kerja Akselerometer

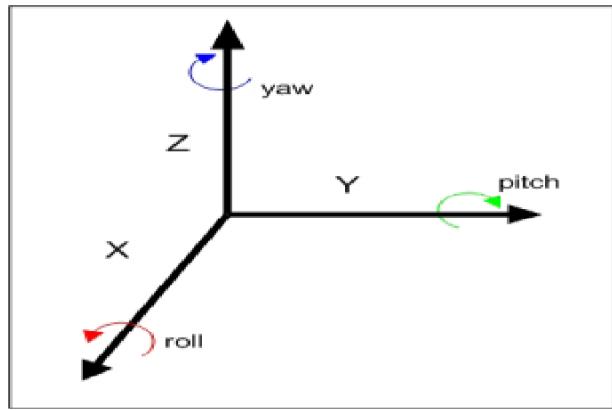
Prinsip kerja sensor ini adalah seperti di tunjukkan oleh gambar 2.12. Sensor menghasilkan panas yang mana jika dimiringkan maka

panas tersebut akan memanasi bagian pinggir sensor yang merupakan bahan seperti thermocoupler. sehingga jika dipanaskan akan menghasilkan perubahan dan perbedaan panas yang mengakibatkan terjadinya perubahan tegangan yang nantinya dimasukkan kedalam ADC.



Gambar 2.12 Perinsipkerja akselerometer

Sensor akselerometer 3 sumbu ini memiliki sifat pergerakan *roll*, *pitch* dan *yaw*. Sifat ini merupakan sifat pergerakan terhadap sumbu X, Y dan Z yang telah disesuaikan dengan titik gravitasi bumi. Sumbu X merepresentasikan gerakan perputaran. Sumbu merepresentasikan gerakan terhadap kemiringan pada sumbu Y .Sumbu Z merepresentasikan gerakan menoleh atau perbelokan terhadap bidang. Gambar 2.13 menunjukan analogi pergerakan pada sensor akselerometer 3 sumbu.



Gambar 2.13. Analogi pergerakan sensor akselerometer 3 sumbu

b. Accelerometer H48C

H48C merupakan sensor percepatan yang mampu mendeteksi pergerakan dari 3 sumbu yaitu x, y, dan z. Sensor ini memberikan keluaran berupa data digital hasil konversi tegangan dengan resolusi ADC 12 bit. Percepatan tiap sumbu(G)³ dapat dihitung dengan persamaan berikut :

$$G = (Axis - vRef) \times 0,0022$$

Dimana : G = Percepatan

Axis = Tegangan keluaran tiap sumbu

vRef = Tegangan referensi ADC

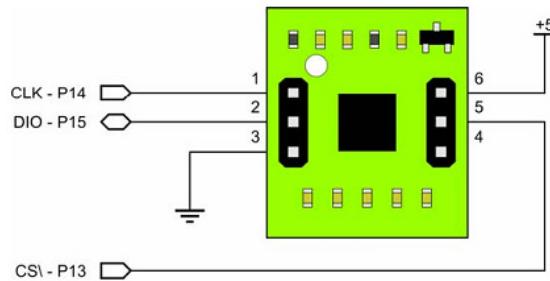
Skematik konfigurasi sederhana dari H48C dapat dilihat pada gambar 2.14.

Pada gambar 2.14, fungsi pin CLK adalah sebagai *synchronous clock input* karena penggunaan bersama jalur data (DIO)⁴. Sedangkan

³Simbol G diambil dari *g-force*

⁴DIO = bi-directional digital input/output dimana modul akan mengirimkan data dan meminta data referensi dari jalur ini.

CS merupakan pin *chip select* yang digunakan untuk memilih modul yang digunakan apabila modul yang dipakai lebih dari satu.



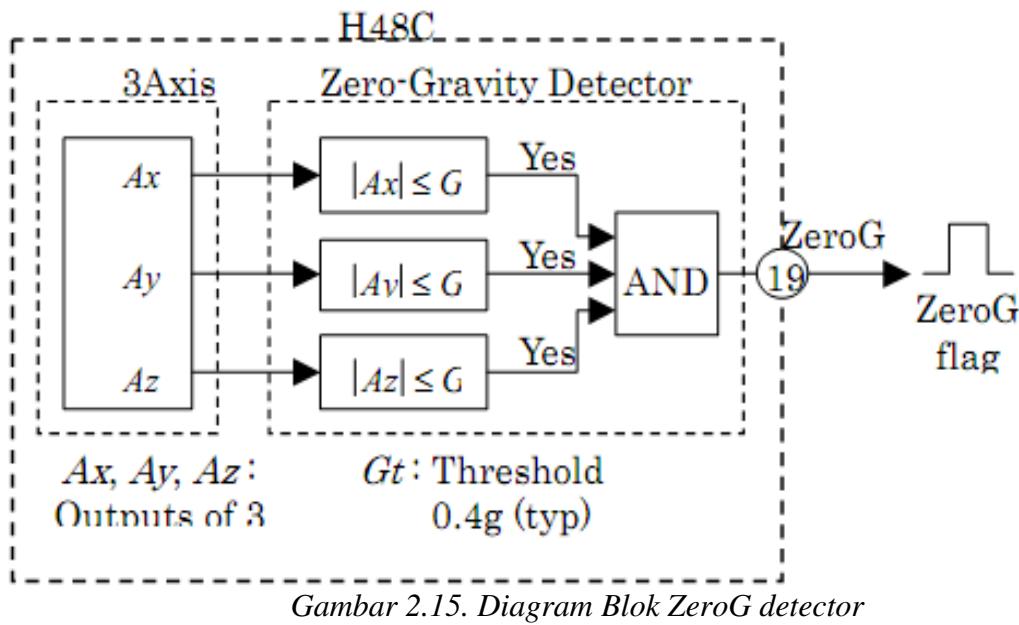
Gambar 2.14. Skematik konfigurasi sederhana dari H48C

Dalam penelitian ini modul sensor H48C yang digunakan adalah modul *3 axis accelerometer* H48C buatan Parallax yang mana sensor percepatan H48C telah terintegrasi dengan *analog signal conditioning* MCP3204.

Prinsip pengiriman dan penerimaan data dari H48C adalah dengan menggunakan teknik geser keluar (*shift out*) untuk mengirim data, dan teknik geser kadalam (*shift in*) untuk menerima data dari kontroler (*host*).

Pada H48C terdapat *zeroG* detector untuk kalibrasi perhitungan. Pada kondisi jatuh bebas, output *zeroG* bernilai 3.3V.

Gambar 2.15 menunjukkan diagram blok dari *zeroG detector*.



Gambar 2.15. Diagram Blok ZeroG detector

c. Pengukuran Perpindahan Posisi

Dalam domain waktu (t), jarak perpindahan sesaat (s_t) dapat diperoleh dengan melakukan integrasi data kelajuan sesaat (v_t) dan kelajuan sesaat (v_t) dapat diperoleh dengan melakukan integrasi percepatan sesaat (a_t). Persamaan untuk menghitung jarak perpindahan sesaat (s_t) dari kelajuan sesaat (v_t) adalah sebagai berikut (persamaan 2.5).

Sedangkan untuk menghitung kelajuan sesaat (v_t) dari percepatan sesaat (a_t) adalah sesuai persamaan 2.6 berikut.

$$v(t) = \int_{t_a}^{t_b} a(t) dt \dots \quad (2.6)$$

Berdasarkan persamaan 2.6 dan persamaan 2.7 untuk memperoleh jarak perpindahan sesaat (s_t) dari percepatan sesaat (a_t)

dapat dilakukan substitusi persamaan 2.7 ke persamaan 2.6 sehingga menjadi persamaan 2.7.

$$s(t) = \int_{t_a}^{t_b} a(t) dt \dots \quad (2.7)$$

Dimana : t = waktu

$s(t)$ = Jarak perpindahan setelah t

$a(t)$ = percepatan sesaat pada waktu t

t_a = Waktu awal

t_b = Waktu akhir

II.4 Sistem Komunikasi Nirkabel (wireless) RF (Radio Frekuensi)

Sistem komunikasi nirkabel merupakan suatu sistem komunikasi yang menggunakan media transmisi non fisik (kabel tembaaga, kabel fiber). Umumnya, sistem komunikasi ini memanfaatkan gelombang elektromagnetik sebagai media penyalurnya. Dengan kata lain, sebuah data / informasi dikirimkan dengan cara dimodulasi kedalam gelombang elektromagnetik. Pada sistem komunikasi nirkabel menggunakan RF berarti media transmisi yang digunakan yakni gelombang elektromagnetik dalam *bandwidth* gelombang radio yakni pada rentang frekuensi 300 Hz sampai dengan 300 GHz.

Karena pengiriman informasi menggunakan frekuensi radio, maka komunikasi ini diatur oleh jenis hukum yang sama dan digunakan untuk mengatur hal-hal seperti AM/FM radio. Federal Communications Commission (FCC) mengatur penggunaan alat dari wireless LAN. Dalam pemasaran wireless sekarang, menerima beberapa standard operasional dan syarat dalam

Amerika Serikat yang diciptakan dan dirawat oleh *Institute of Electrical Electronic Engineers (IEEE)*. Beberapa Standar wireless LAN :

IEEE 802.11 – Standar asli wireless LAN menetapkan tingkat perpindahan data yang paling lambat dalam teknologi transmisi light-based dan RF.

IEEE 802.11b – Menggambarkan tentang beberapa transfer data yang lebih cepat dan lebih bersifat terbatas dalam lingkup teknologi transmisi.

IEEE 802.11a – Gambaran tentang pengiriman data lebih cepat dibandingkan (tetapi kurang sesuai dengan) IEEE 802.11b, dan menggunakan 5 GHZ frekuensi band UNII.

IEEE 802.11g – Syarat yang paling terbaru berdasar pada 802.11 standard yang menguraikan transfer data sama dengan cepatnya seperti IEEE 802.11a, dan sesuai dengan 802.11b yang memungkinkan untuk lebih murah.

II.4.1 Protokol

Protokol adalah sebuah standar yang mengatur atau mengijinkan terjadinya hubungan komunikasi, dan perpindahan data antara dua atau lebih perangkat. Protokol dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

Hal – Hal yang perlu diperhatikan mengenai protokol adalah :

1. Melakukan deteksi ada atau tidak adanya koneksi fisik antar perangkat keras.
2. Melakukan metode jabat tangan (*Handshaking*).
3. Negosiasi berbagai macam karakteristik hubungan.
4. Bagaimana mengawali dan mengahiri suatu pesan.
5. Bagaimana format pesan yang digunakan.
6. Yang harus dilakukan saat terjadi kerusakan pesan atau pesan tidak sempurna.
7. Mendeteksi rugi – rugi pada hubungan jaringan dan langkah – langkah yang dilakukan selanjutnya.
8. Mengakhiri suatu koneksi

Dalam membuat protokol ada tiga hal yang harus dipertimbangkan yaitu efektifitas, kehandalan, dan fleksibilitas.

II.4.2 Modul X-BeePro

Pada masa sekarang ini telah banyak dikembangkan modul *wireless RF*. Salah satu modul *wireless RF* yang sering dipakai adalah *X-Bee Pro* yang dibuat oleh *Maxstream*. *X-Bee pro* dirancang agar dapat memenuhi teknologi *Zigbee/IEEE 802.15.4*.

Zigbee/IEEE 802.15.4 merupakan teknologi yang memfokuskan transfer data (*data rate*) rendah, konsumsi daya rendah, biaya murah dan target protokol untuk jaringan *wireless* aplikasi otomasi dan kendali *remote*. Modul *X-Bee pro* yang digunakan memiliki spesifikasi XBP-

24/1083 yang beroperasi pada daerah 2,4 GHz. Fitur yang dimiliki oleh modul ini adalah :

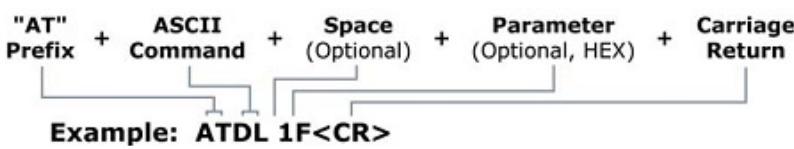
1. Jarak komunikasi *indoor* sampai 300 m dan *outdoor* hingga 1500 m LOS (*Line Of Sight*)
2. Sensitivitas penerimaan -100 dBm.
3. RF data rate 250.000 bps.
4. Setiap *channel* menyediakan alamat jaringan lebih dari 65.000 alamat.
5. Mendukung topologi *peer to peer*, *point to multiple point* dan *point to point*.
6. Bentuk paket modul relatif kecil.
7. Kompatible dengan perangkat lain yang mendukung teknologi Zigbee/IEEE 802.15.4.

Xbee *pro* menyediakan beberapa mode pengalamatan untuk proses komunikasi. Salah satu mode pengalamatan yang disediakan adalah *short 16 bit addressing*. Mode pengalamatan ini memiliki beberapa parameter yaitu :

1. MY, merupakan alamat diri dari setiap modul.
2. DL, merupakan alamat tujuan komunikasi.
3. CH, merupakan channel dimana komunikasi RF terjalin
4. ID, merupakan alamat PAN (Personal Area Networking) ID dari tiap modul RF.

Pengaturan parameter pada modul wireless RF dilakukan dengan menggunakan AT Command. Beberapa hal yang perlu diperhatikan dalam penggunaan AT Command adalah :

1. Untuk membuka AT Command mode kirim 3 character plus ("++") dalam waktu kurang dari 1 detik.
2. Untuk mengirim AT Command gunakan aturan sebagai berikut (gambar 2.16).



Gambar 2.16 Aturan penulisan AT Command

3. Untuk pembacaan parameter biarkan parameter kosong
4. Jika AT Command sukses dikirimkan dan dieksekusi maka akan ada respon OK (untuk pengaturan) atau nilai parameter (untuk pembacaan)
5. Untuk menyimpan parameter konfigurasi kirim ATWR
6. Untuk menutup AT Command mode kirim ATCN.

II.4.3 Enkripsi / Deskripsi Data

Enkripsi / deskripsi data dimaksudkan untuk menjaga keamanan data selama proses transmisi data. Metode yang dipakai pada enkripsi / deskripsi data adalah metode Caesar. Prinsip utama dari metode Caesar adalah adanya suatu pergeseran dari elemen data yang akan dienkripsi / dideskripsi. Pada sistem enkripsi yang dibuat nilai pergeseran ditentukan oleh suatu konstanta ditambah nilai BCD

dari elemen FCS yang kedua. Elemen dari protokol yang dienkripsi / dideskripsi adalah setelah delimiter "@" sampai data yang dikirimkan, untuk code data, FCS dan terminator tidak dienkripsi. Secara umum prosesenkripsi dapat diilustrasikan sebagai berikut (gambar 2.17).

@	1	2	N	D	O	K	R	3	1	*
64	49	50	78	68	79	75	82	51	49	42
↓										
64	60	61	89	79	90	86	82	51	49	42

@	<	=	Y	O	Z	V	R	3	1	*
@	<	=	Y	O	Z	V	R	3	1	*

Gambar 2.17 Ilustrasi metode Caesar Chipper

Dari gambar di atas dapat dilihat bahwa ketika sebuah frame protokol dikirimkan, FCS ditempatkan setelah kode data untuk mengecek apakah ada data yang hilang atau tidak sesuai. Secara umum FCS merupakan hasil dari XOR kode ASCII karakter – karakter sebelumnya yang diimplementasikan ke dalam 2 byte BCD dari nilai XOR dari kode ASCII karakter tersebut. Jadi secara umum FCS juga digunakan untuk pengecekan data yang diterima selain adanya pengecekan delimiter "@" dan terminator "*" sebelumnya.

Secara umum ilustrasi dari perhitungan FCS adalah sebagai berikut.

@	1	2	N	D	O	K	R	3	1	*
FCS Range Calculation								FCS	T	

Selanjutnya setiap karakter pada frame dikonversi menjadi kode ASCII dan di-XOR-kan dengan kode ASCII karakter sesudahnya sehingga diperoleh suatu nilai ASCII terakhir dan FCS merupakan 2 byte nilai BCD dari kode ASCII tersebut.

II.5 Mikrokontroller

Mikrokontroler merupakan komponen utama yang berfungsi untuk mengolah informasi yang diperoleh oleh, kemudian diproses untuk mengambil kesimpulan respon apa yang akan dilakukan oleh sistem. Mikrokontroler memiliki sifat yang dapat diprogram oleh pemakai sehingga sangat fleksibel jika dimanfaatkan sebagai piranti kontrol.

Sebuah mikrokontroler memiliki minimal tiga blok penting yakni :

1. CPU (*Central Processing Unit*)

Yakni unit dimana terjadinya operasi aritmatika ataupun logika serta register-register geser. Semua perhitungan sistem kontrol dan operasi input/output terjadi di bagian ini.

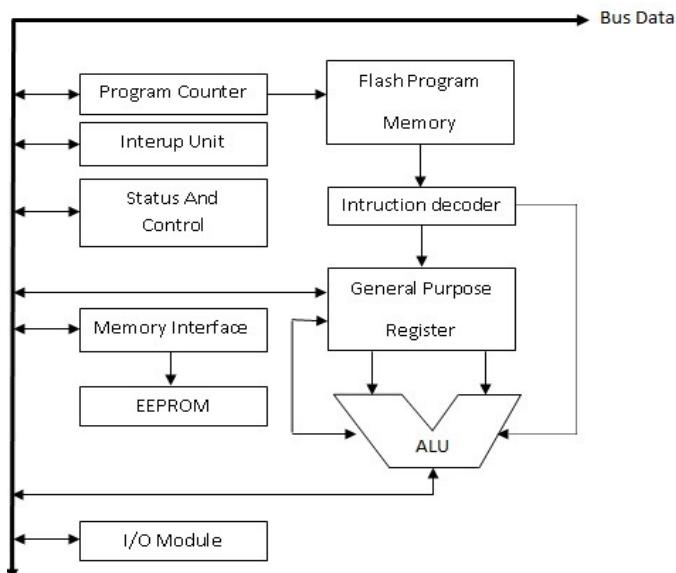
2. *Memori*

Merupakan unit penyimpanan data. Instruksi-instruksi yang paling dasar di set pada *high level* program, yang terinstal dalam *Read Only Memory (ROM)*. Sedangkan program-program *logic* disimpan pada *Electrically Eraseble Permanent Read Only Memory (EEPROM)*. Selama program tersimpan dalam EEPROM, maka program logika tersebut masih dapat dilakukan perubahan sesuai dengan kebutuhan.

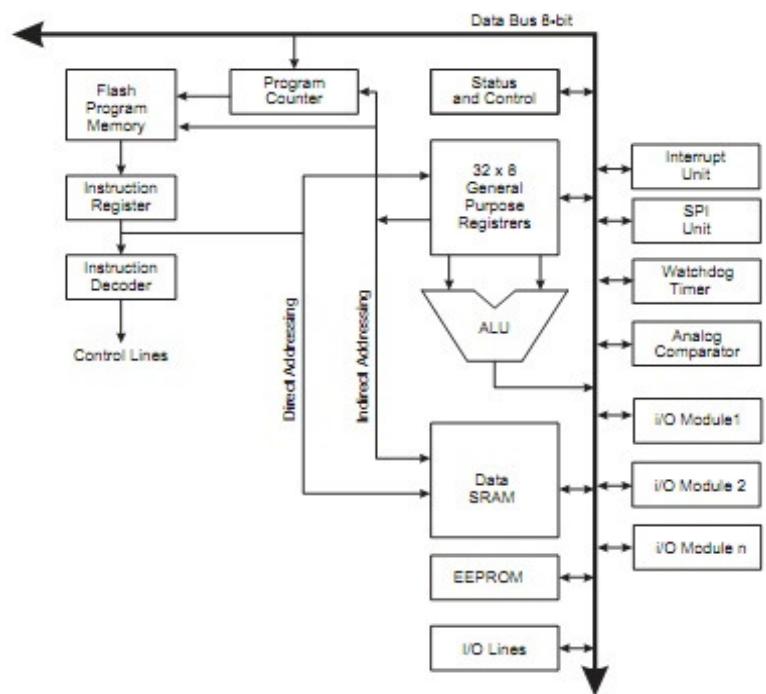
3. I/O Unit (*Unit Input / Output*)

Merupakan bagian yang bertugas melakukan komunikasi dengan dunia luar. Unit input bertugas dalam memproses masukan dari dunia luar untuk diproses lebih lanjut oleh CPU. Sedangkan unit output merupakan bagian yang bertugas sebagai perantara CPU melakukan operasi terhadap piranti luar yang merupakan unit aktuator dari sebuah sistem ataupun unit HMI (*Human and Machine Interfaces*) seperti LCD *display*.

Gambar 2.18 merupakan blok diagram dari arsitektur mikrokontroler sederhana. Akan tetapi secara garis besar, blok diagram tersebut dapat mewakili sebagian besar jenis mikrokontroler karena pengembangan mikrokontroler didasarkan dari blok diagram tersebut. Sebagai contoh ATmega8 (ditunjukkan oleh gambar 2.19) yang merupakan mikrokontroler jenis AVR buatan dari ATMEL Corp Inc.



Gambar 2.18 Blok diagram arsitektur dasar mikrokontroler



Gambar 2.19 Arsitektur mikrokontroler ATmega8

BAB III

METODE PENELITIAN

III.1 Waktu dan Tempat

Penelitian ini akan dilaksanakan selama 5 bulan yang akan dilaksanakan mulai dari bulan Februari 2012 sampai dengan Mei 2012. Sedangkan penelitian akan dilaksanakan di Laboratorium Teknik Elektro Universitas Jenderal Soedirman Purwokerto.

III.2 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Seperangkat komputer dengan sistem operasi Windows 7, RAM 1Gb
2. Arduino *Board* dan *Software Arduino IDE 0022*.
3. Paket *mote (Xbee Pro Series I)*.
4. Unit RTC (IC DS1307, *Crystal 32.768KHz*, Baterai 3V).
5. Model fisik pergerakan tanah.

III.3 Metode Penelitian

Metode penelitian yang digunakan adalah sebagai berikut.

III.1 Studi Pustaka

Pada metode ini dilakukan dengan cara mencari dan membaca buku-buku referensi ataupun juga dari sumber-sumber internet yang menunjang untuk penelitian ini.

III.2 Identifikasi Permasalahan

Tahap ini mencakup identifikasi permasalahan dilapangan sehingga dapat menentukan spesifikasi peralatan yang sesuai dengan kondisi lapangan. Pada tahap ini, dilakukan juga pencatatan daftar kebutuhan sebagai persiapan dari perancangan.

III.3 Perancangan Sistem

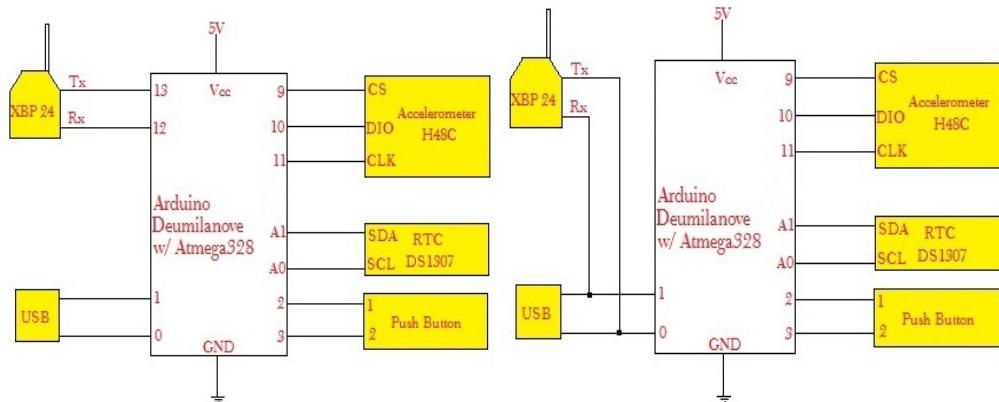
Tahap perancangan ini mengacu terhadap tahap sebelumnya yakni identifikasi permasalahan sehingga tahap ini merupakan tahap merancang suatu alat yang sesuai dengan kebutuhan.

Dalam tahap ini dilakukan proses – proses berikut :

a. Desain peralatan

Mencakup desain sensor dan pengkondisian sinyal, kontroler, unit komunikasi serta unit catu daya.

Secara umum, unit elektronis ditunjukan oleh diagram blok pada gambar 3.1. Terdapat perbedaan konfigurasi antara *gateway* dan *router* dikarenakan kebutuhan jalur komunikasi yang berbeda.



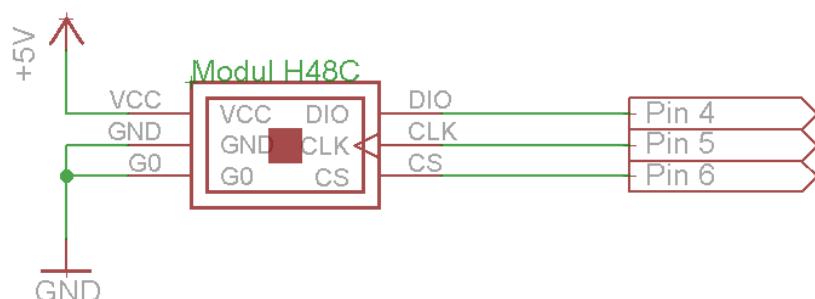
Gambar 3.1. Diagram Blok unit elektronik(kiri:gateway, kanan:router)

1. Desain sensor dan Pengkondisian sinyal

Sensor yang digunakan berupa *accelerometer* H48C untuk mendeteksi pergerakan tanah.

Dalam perancangan rangkaian , sensor *accelerometer* H48C telah terintegrasi dengan IC MCP324 sehingga keluaran modul berupa data serial sehingga tidak diperlukan lagi rangkaian pengkondisian sinyal yang lain.

Skematik pengkondisian sinyal modul *accelerometer* H48C sesuai gambar 3.2 berikut:



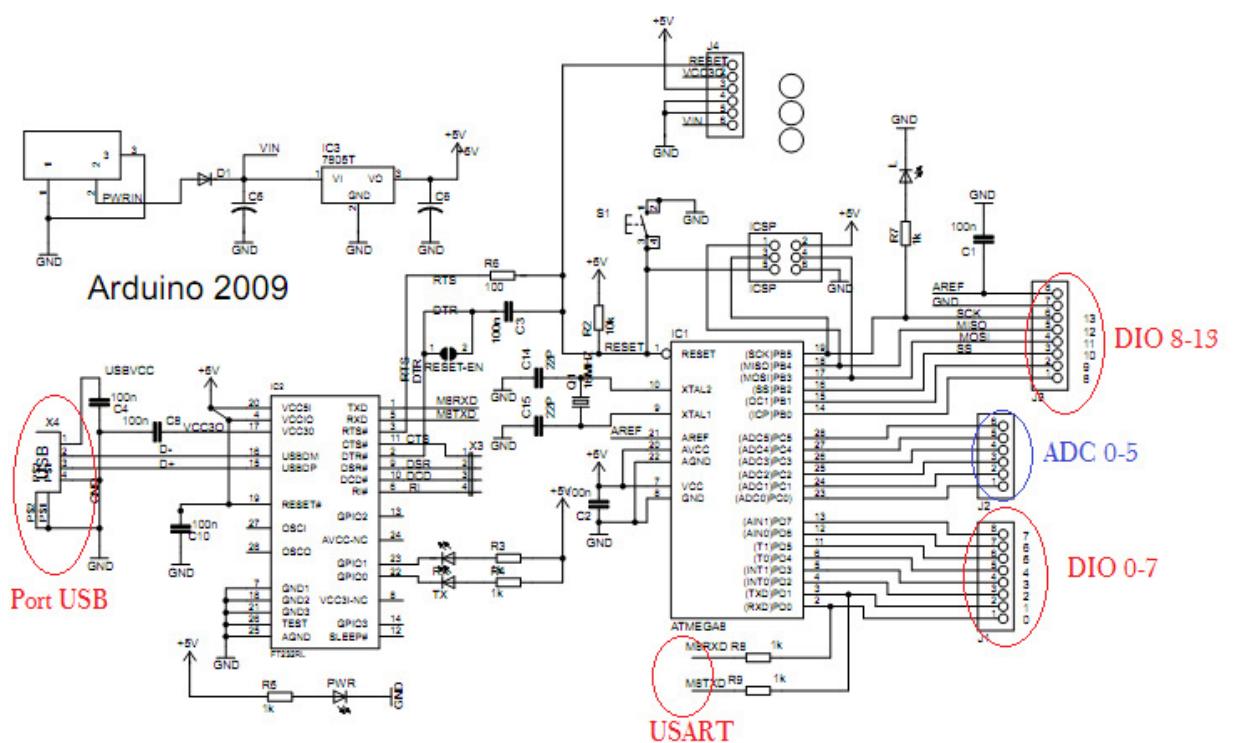
Gambar 3.2. Pengkondisian Sinyal pada accelerometer H48C

2. Kontroler

Kontroler yang digunakan merupakan kit mikrokontroler *Arduino board* tipe *Deumilanove* dengan mikrokontroler Atmega328.

Pemilihan penggunaan *Arduino Board* dikarenakan kemudahan dalam penggunaan terutama fasilitas *plug and play* pada komunikasi serial dengan komputer.

Gambar 3.3 menunjukkan skematik *arduino board* tipe *Deumilanove*. Dapat dilihat bahwa *Arduino* board memiliki 13 jalur input/output digital, 5 jalur input analog dan port USB untuk komunikasi serial dengan komputer via USB. Untuk komunikasi dengan perangkat lain dapat menggunakan jalur Tx dan Rx komunikasi USART.



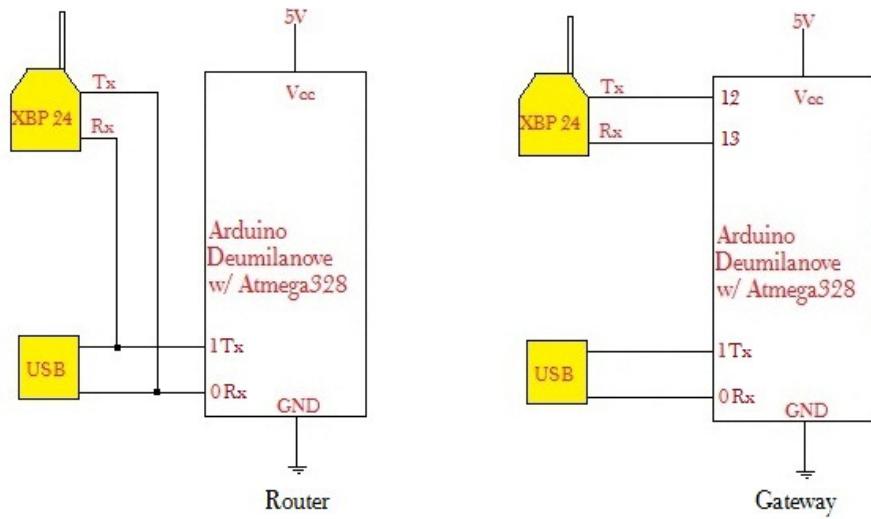
Gambar 3.3. Skematik *Arduino Deumilanove*

3. Unit Komunikasi

Unit komunikasi antar *node* menggunakan modul *wireless Xbee Pro 24 Series 1*. Pada perancangan, topologi yang digunakan adalah topologi *star* dengan konfigurasi 2 buah *node* sebagai *router* dan 1 buah *node* sebagai *gateway / base station*. Jalur komunikasi *gateway* dengan komputer server menggunakan USB, sehingga terdapat perbedaan konfigurasi pin pada mikrokontroler antara *router* dan *gateway*.

Pada *router*, pin Tx dari modul Xbee dihubungkan dengan pin Rx dari *arduino board* dan pin Rx dari modul Xbee dihubungkan dengan pin Tx dari *arduino board*. Sedangkan pada *gateway*, pin Tx modul Xbee dihubungkan dengan pin 12 dari *arduino board* dan pin Rx modul Xbee dihubungkan dengan pin 13 dari *arduino board*. Hal ini dikarenakan pin Tx dan Rx pada *arduino board* yang dikonfigurasikan sebagai *gateway* digunakan untuk komunikasi dengan komputer server via USB.

Gambar 3.4 menunjukkan konfigurasi pin modul Xbee dengan *arduino board*.



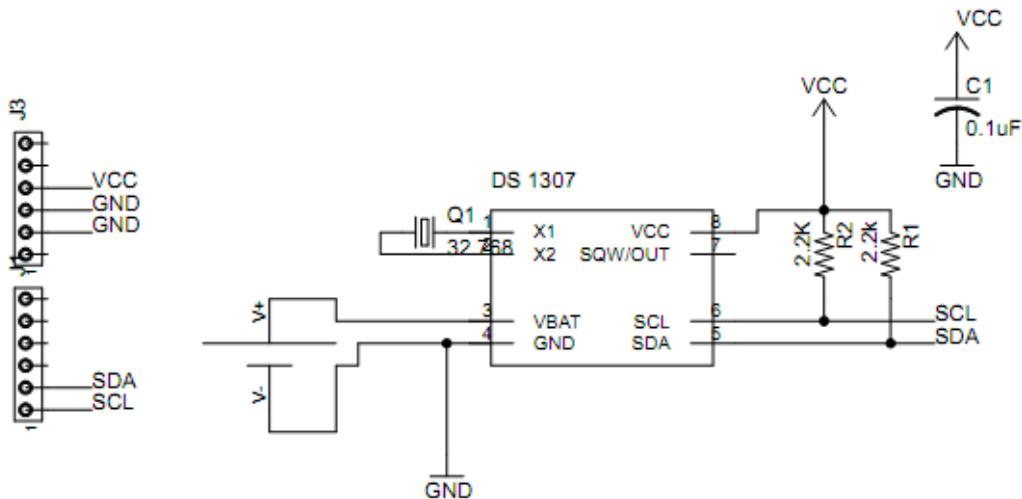
Gambar 3.4. Konfigurasi Pin pada modul Xbee dengan Arduino board

4. Unit RTC (*Real Time Condition*)

Unit RTC digunakan sebagai unit *sinkronisasi* waktu dari tiap - tiap node. Unit RTC terhubung dengan *arduino board* pada pin SCL/A0 dan Pin SDA/A1 karena menggunakan konfigurasi *2 wire*.

Piranti yang digunakan adalah IC DS1307. Tegangan kerja unit RTC diberikan oleh *arduino board* dan menggunakan baterai *lithium* 3V sebagai *back up voltage* jika terjadi kegagalan tegangan suplai dari *arduino board*.

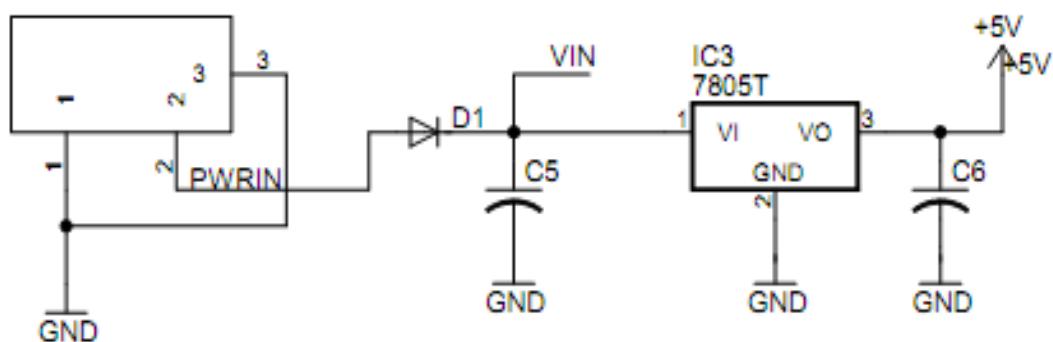
Skematik unit RTC ditunjukan oleh gambar 3.5.



Gambar 3.5. Skematik rangkaian RTC menggunakan DS1307

5. Unit Catu Daya

Setiap node disuplai oleh baterai karena setiap *node* diletakan diberbagai tempat dengan jarak yang relatif jauh sehingga penggunaan baterai pada tiap *node* akan lebih efektif. Penyesuaian level tegangan baterai dengan tegangan kerja *node* menggunakan rangkaian regulator dengan memanfaatkan IC regulator 7805. Skematik rangkaian regulator 7805 adalah sebagai berikut (gambar 3.6) :



Gambar 3.6. Skematik regulator tegangan menggunakan IC 7805

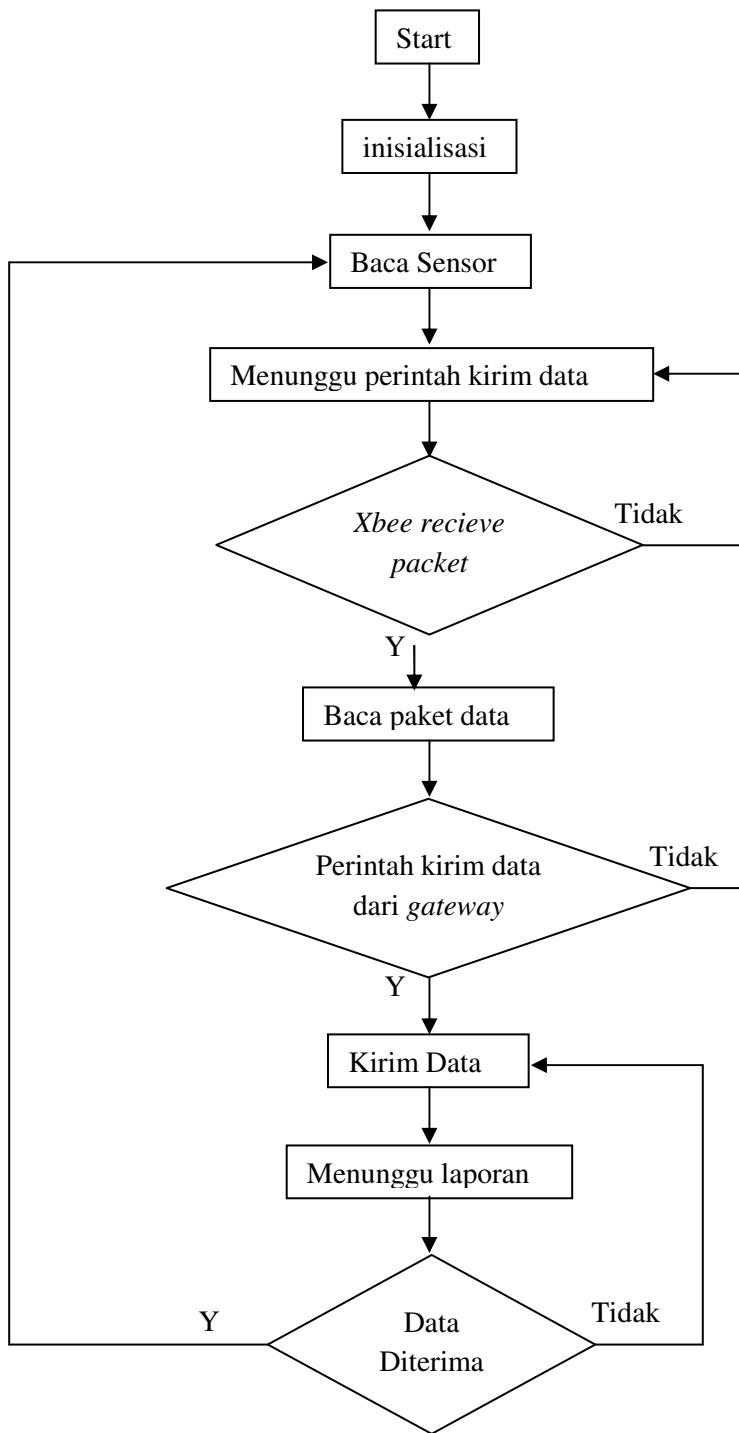
b. Perancangan perangkat lunak (software)

Perangkat lunak terdiri dari program untuk *node* dan program untuk server. Program untuk *node* sendiri terdiri dari 2 jenis yaitu program untuk *router* dan program untuk *gateway*.

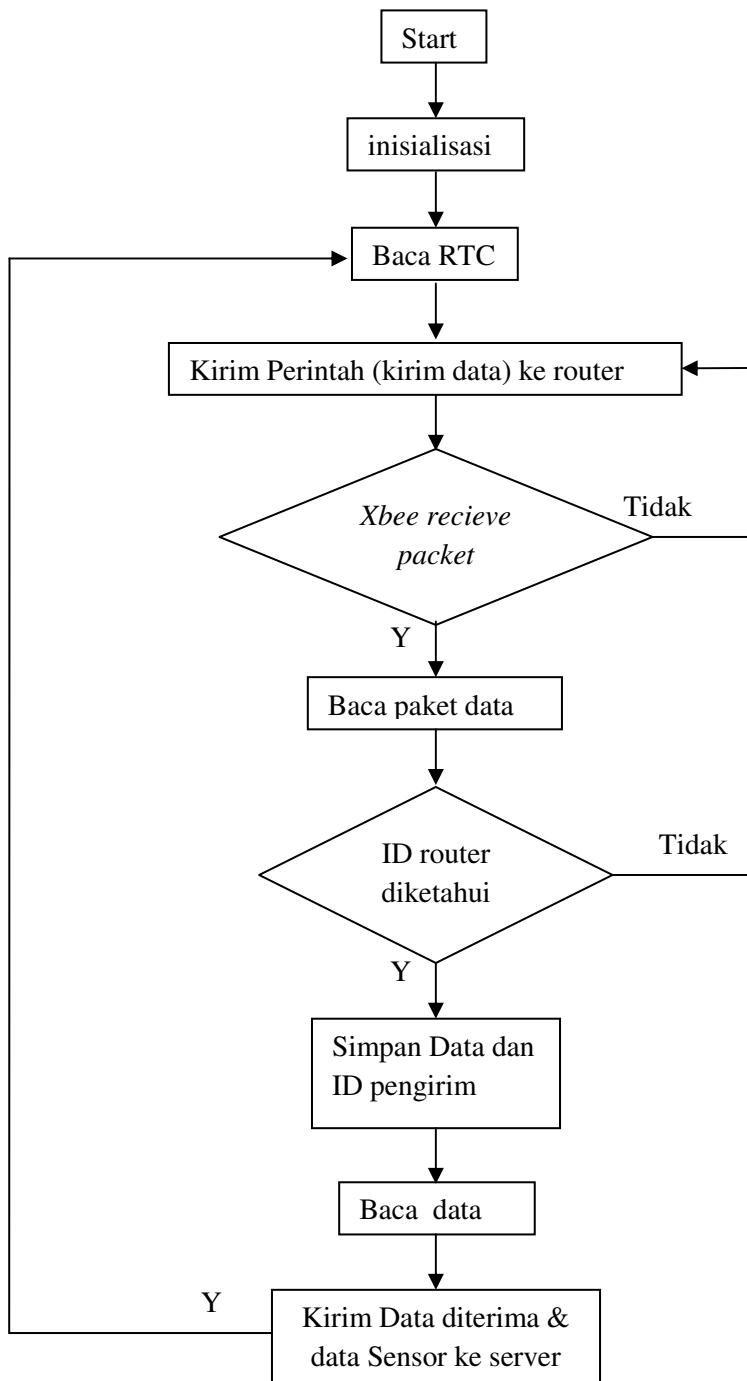
Flowchart program pada router sesuai gambar 3.7. *Flowchart* untuk program pada gateway sesuai gambar 3.8.

Sedangkan *flowchart* program pada server merupakan program *data logger* yang berfungsi untuk mengumpulkan informasi yang dikirimkan oleh *gateway* yang kemudian disimpan kememori dengan format data *comma separated vector*(.csv) untuk mempermudah proses pengolahan data selanjutnya.

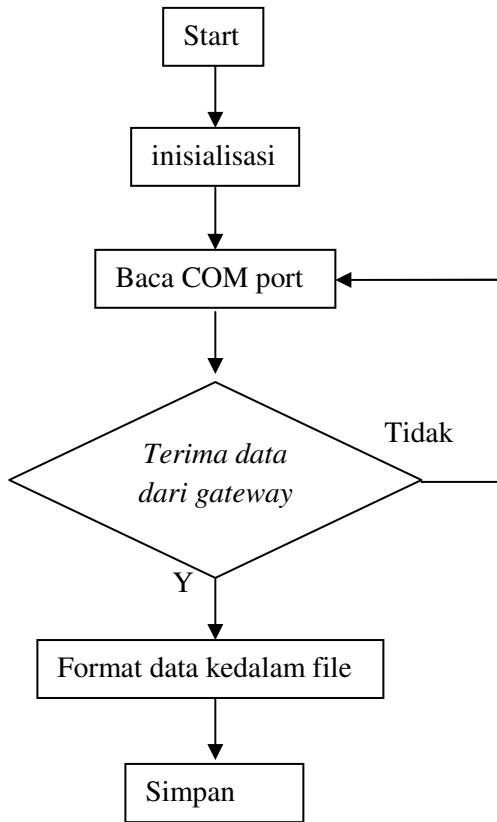
Flowchart program pada server ditunjukan oleh gambar 3.9



Gambar 3.7.Flowchart program pada router



Gambar 3.8. Flowchart program gateway



Gambar 3.9. Flowchart program pada komputer server

III.4 Pembuatan Alat

Berpedoman dari hasil perancangan pada tahap sebelumnya, maka tahap ini mencakup beberapa proses yakni :

- Pembuatan sistem elektronis dan mekanis
Mencakup pembuatan sistem minimum mikrokontroler, pembuatan sensor – sensor, pembuatan unit pengkondisian sinyal, unit komunikasi (*wireless*) serta unit catu daya.
- Pembuatan perangkat lunak (*software*)

Merupakan pengaplikasian dari diagram alir program yang dibuat dengan menggunakan bahasa pemrograman.

c. Pembuatan perangkat keras (*hardware*)

Merupakan pembuatan perangkat keras dengan berpedoman dari desain perangkat keras yang didapat dari proses sebelumnya.

III.5 Pengujian Sistem

Proses pengujian sistem dilakukan di laboratorium. Pengujian ini dilakukan untuk dapat memantau kinerja sistem, kesesuaian kinerja sistem dengan apa yang diharapkan (hasil identifikasi permasalahan), serta kehandalan sistem.

Proses yang dilakukan dalam pengujian sistem diantaranya:

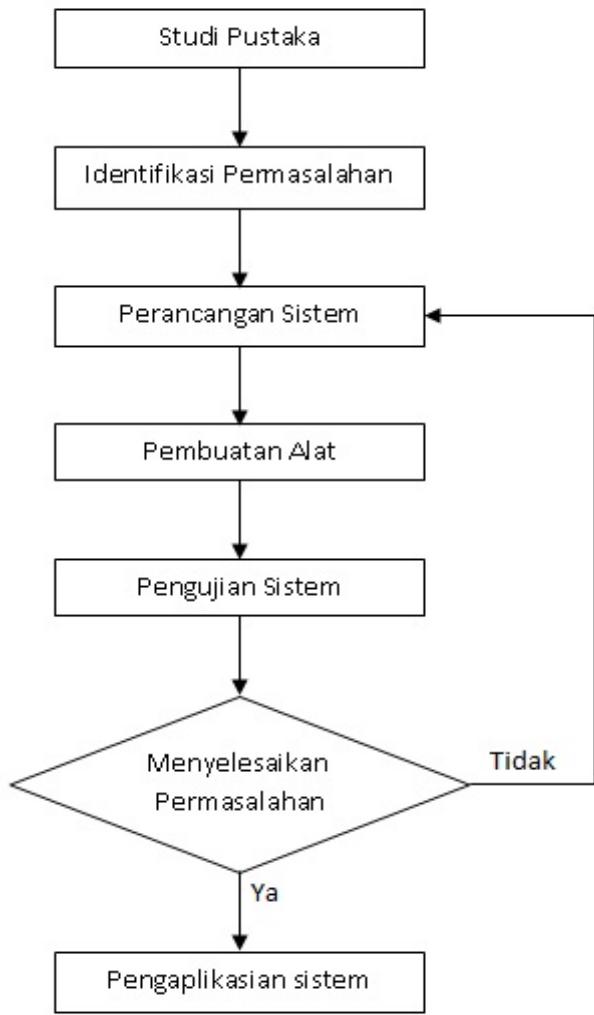
- a. Kalibrasi *accelerometer* H48C dalam melakukan pengukuran percepatan pergerakan tanah.
- b. Menentukan frekuensi optimal pembacaan data percepatan menggunakan *accelerometer* H48C dan pengiriman data menggunakan Xbee Pro Series 1.
- c. Membandingkan kuat sinyal komunikasi dalam berbagai medan.
- d. Menghitung konsumsi daya tiap *mote*.

Jika pada tahap pengujian ini ditemui hal – hal yang tidak sesuai dengan apa yang diaharapkan maka dapat dilakukan pengkajian ulang terhadap proses perancangan sistem.

III.6 Pengaplikasian Sistem

Proses ini dilakukan jika pengujian sistem telah berhasil sehingga dianggap sistem yang dibuat telah siap diaplikasikan dilapangan.

Secara keseluruhan metode penelitian yang digunakan dapat dilihat dari *flowchart* (diagram alir) gambar 3.10 berikut :



Gambar 3.10. Diagram alir metode penelitian

III.4 Jadwal Penelitian

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Bulan											
		Febuari 2012				Maret 2012				April 2012			
		I	II	III	IV	I	II	III	IV	I	II	III	IV
1	Studi pustaka												
2	Pembuatan proposal												
3	Perancangan sistem												
4	Persiapan alat dan bahan												
5	Pembuatan sistem												
6	Pengujian sistem												
7	Pengaplikasian sistem												
8	Pembuatan laporan												

BAB IV

HASIL DAN PEMBAHASAN

IV.1 Unit RTC (Real Time Clock)

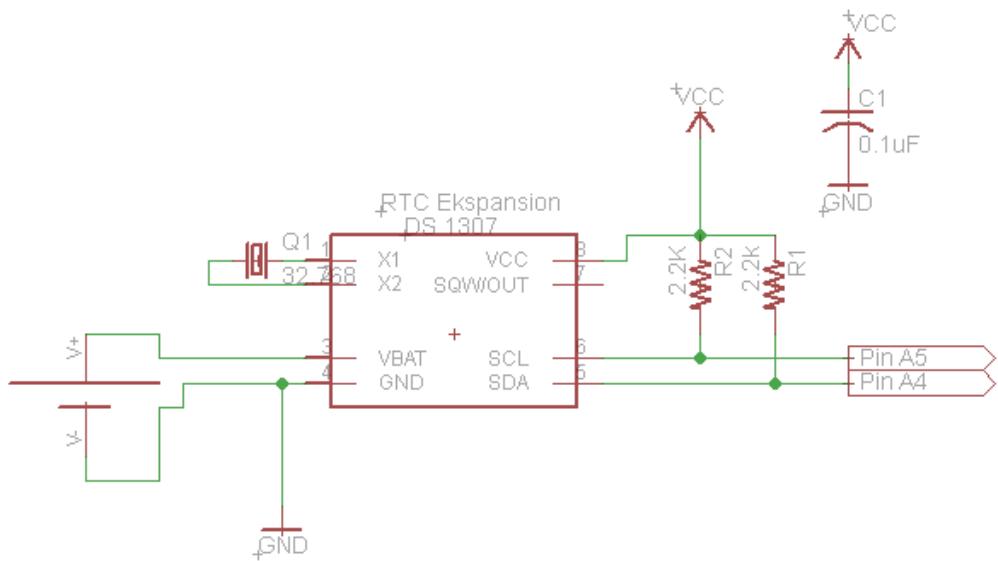
Unit RTC digunakan untuk mengirimkan data waktu pembacaan sensor. Hal ini diperlukan untuk mengetahui waktu pembacaan data sensor, dan koreksi terhadap adanya paket data yang hilang.

IV.1.1 Skematik Rangkaian RTC

DS1307 merupakan IC RTC yang menggunakan protokol I²C (*Inter IC Bus*) yang sesuai dengan protokol TWI (*2 wire serial interfaces*) pada mikrokontroler AVR. Dalam komunikasi menggunakan protokol I²C hanya diperlukan 2 jalur data yakni SCL (*Serial Clock*) dan SDA (*Serial Data*).

Arduino board yang digunakan adalah deumilanove dengan mikrokontrolernya adalah Atmega328 yang mana pin SCL terletak pada kaki no 28 pada Atmega328 dan dalam *arduino board* disebut sebagai pin A5, sedangkan pin SDA terletak pada kaki no 27 pada Atmega328 dan dalam *arduino board* disebut sebagai pin A4.

Skematik rangkaian percobaan untuk modul RTC ditunjukkan oleh gambar 4.1.



Gambar 4.1 Skematik rangkaian modul RTC dengan IC DS1307

IV.1.2 Algoritma Penggunaan RTC (*Real Time Clock*) dengan IC DS1307

Pada penggunaan IC DS1307 dilakukan 2 proses yakni menulis data ke IC DS1307 (dalam hal ini adalah seting waktu) dan yang kedua adalah membaca data waktu dari IC DS1307. Prosedur kedua proses ini mengacu pada protokol I²C agar komunikasi berjalan sebagai mana mestinya.

Sesuai dengan protokol I²C, prosedur penulisan data dilakukan dengan langkah – langkah sebagai berikut.

1. *Master* mengirim *start sequence*
2. *Master* mengirim alamat *slave* yang akan ditulis data dengan bit R/W low (alamat genap)
3. *Master* mengirim alamat dari internal register yang ingin ditulis
4. *Master* mengirim byte data

5. Jika data belum tertulis semua, setelah *slave* mengirim ACK=0

maka byte data berikutnya dikirim

6. *Master* mengirim *stop sequence*.

Sedangkan prosedur pembacaan data dilakukan dengan langkah

– langkah sebagai berikut.

1. *Master* mengirim *start sequence*

2. *Master* mengirim alamat *slave* yang akan ditulis data dengan
bit R/W low (alamat genap)

3. *Master* mengirim alamat dari internal register yang ingin dibaca

4. *Master* mengirim *start sequence*

5. *Master* mengirim alamat *slave* yang akan ditulis data dengan
bit R/W high (alamat ganjil)

6. *Master* membaca byte data dari *slave*

7. *Master* mengirim *stop sequence*.

Dengan konfigurasi *Arduino board* sebagai *master* dan modul
RTC sebagai *slave* maka prosedur penulisan data dan pembacaan data
sesuai protokol I2C diatas dapat disusun dengan notasi algoritmik
berikut.

```
{-procedure menulis data ke DS 1307 untuk set waktu-}
Procedure setDS1307 (input=
second,minute,hour,dayOfWeek,dayOfMonth,Year)
    Set second as BCD {-ubah ke BCD format--}
    Set minute as BCD
    Set hour as BCD
    Set dayOfWeek as BCD
    Set dayOfMonth as BCD
    Set Year as BCD
    Set I2C address 0x68      {-alamat DS1307--}
    Send (0)                  {-Start Sequence--}
    Send (0x68)                {-kirim alamat slave-}
```

```

Write (0x00=second) {--tulis second @00h--}
Write (0x01=minute) {--tulis minute @01h--}
Write (0x02=hour)   {--tulis hour @ 02h--}
Write (0x03=dayOfWeek) {tulis dayOfWeek @03h}
Write (0x04=dayOfMonth){--tulis dayOfMonth @ 04h--}
Write (0x05=Year)    {--tulis Year @ 05h--}
Send (0)           {--Stop sequence--}

{--end procedure--}
*****
{--procedure membaca data dari DS 1307 --}
Procedure getDS1307(output=
second,minute,hour,dayOfWeek,dayOfMonth,Year)
Send (0)           {--Start Sequence--}
Send (0x68)        {--irim alamat slave--}
Send (0)           {--Start Sequence--}
Send (0x68 + 0x01) {--Read data--}
Second=read(0x00)  {--baca alamat 00h--}
Minute=read(0x01)  {--baca alamat 01h--}
Hour=read(0x02)    {--baca alamat 02h--}
dayOfWeek=read(0x03) {--baca alamat 03h--}
dayOfMonth=read(0x04) {--baca alamat 04h--}
year=read(0x05)    {--baca alamat 05h--}
Second=set second as Dec {-----convert BCD ke
Desimal-----}
Minute=set minute as Dec
Hour=set hour as Dec
dayOfWeek=set dayOfWeek as Dec
dayOfMonth=set dayOfMonth as Dec
year=set year as Dec
send(0)           {--stop sequence--}
{--end procedure--}

```

Didalam software arduino IDE versi 0022 telah tersedia *library* “wire.h” yang berfungsi mempermudah pengguna yang akan melakukan komunikasi *master – slave* menggunakan sistem I²C. *Library* ini mengacau pada aturan komunikasi I²C dengan kata kunci untuk memanggil prosedure dan fungsi dari *library* “wire.h” sebagai berikut.

```

Wire.begin();
Wire.beginTransmission(alamat yang dituju);
Wire.endTransmission();
Wire.requestFrom(alamat + bit read);
Wire.send(data bit dalam BCD);
Wire.receive();
Wire.onReceive();
Wire.onRequest() ;

```

Untuk masing – masing prosedure penulisan dan pembacaan data ditunjukan oleh source code dibawah dengan terlebih dahulu memanggil *library* “wire.h” dan mendeklarasikan alamat DS1307 pada 0x68 di bagian inisialisasi.

```
#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68

//*****menulis data set waktu ke DS1307*****
void setDateDs1307(byte second,byte minute,byte hour,byte dayOfWeek,byte dayOfMonth,byte month, byte year){
    Wire.beginTransmission(DS1307_I2C_ADDRESS);
    Wire.send(0); // start clock dengan mengisi bit 7 = 0
    Wire.send(decToBcd(second));
    Wire.send(decToBcd(minute));
    Wire.send(decToBcd(hour));
    Wire.send(decToBcd(dayOfWeek));
    Wire.send(decToBcd(dayOfMonth));
    Wire.send(decToBcd(month));
    Wire.send(decToBcd(year));
    Wire.endTransmission(); }

//*****mengambil data dari DS1307*****
void getDateDs1307(byte *second,byte *minute,byte *hour,byte *dayOfWeek,byte *dayOfMonth,byte *month,byte *year){
    Wire.beginTransmission(DS1307_I2C_ADDRESS);
    Wire.send(0);
    Wire.endTransmission();
    Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
    *second = bcdToDec(Wire.receive() & 0x7f);
    *minute = bcdToDec(Wire.receive());
    *hour = bcdToDec(Wire.receive() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.receive());
    *dayOfMonth = bcdToDec(Wire.receive());
    *month = bcdToDec(Wire.receive());
    *year = bcdToDec(Wire.receive());
}
```

Penulisan data ke DS1307 dilakukan dengan memanggil prosedur setDateDS1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year) dan ini dilakukan pada saat pertama menggunakan IC DS1307, setelah itu data tersimpan dan perhitungan waktu akan terus berjalan.

Selama suplai tegangan pada DS1307 aktif dan tidak ada kerusakan pada rangkaian, maka data waktu pada DS1307 dapat diambil kapan saja. Apabila suplai tegangan pada DS1307 hilang, maka perlu dilakukan penulisan ulang waktu dan tanggal karena data yang tersimpan diregister internal DS1307 merupakan data terakhir IC ini bekerja secara normal (mendapat suplai tegangan).

Contoh pemanggilan prosedur untuk menulis data ke DS1307 adalah sebagai berikut.

```
void loop(){
    second = 30;
    minute = 7;
    hour = 3;
    dayOfWeek = 4;
    dayOfMonth = 18;
    month = 8;
    year = 11;
    setDateDs1307(second,      minute,      hour,      dayOfWeek,
                  dayOfMonth, month, year);
}
```

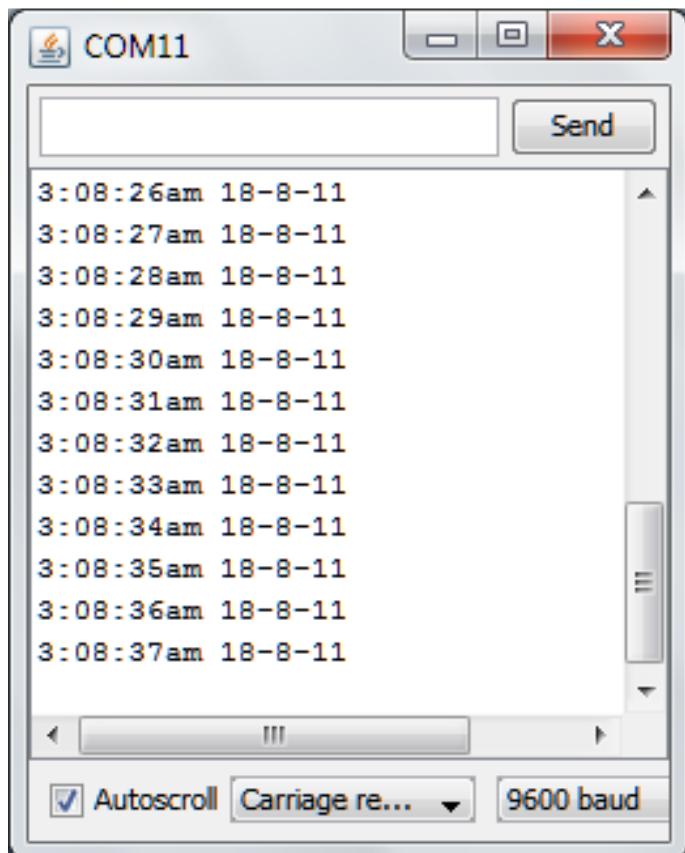
Setelah, data tertulis ke DS1307, langkah selanjutnya adalah pengambilan data kalender pada DS1307. Contoh pemanggilan prosedur untuk membaca data pada DS1307 adalah sebagai berikut.

```

void loop() {
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth,
&month, &year);
if(hour>0){
    if(hour<=12){
        Serial.print(hour, DEC);}
    else {
        Serial.print(hour-12, DEC);} }
else{
    Serial.print("12");}
Serial.print(":");
if (minute < 10) {
    Serial.print("0");}
Serial.print(minute, DEC);
Serial.print(":");
if (second < 10) {
    Serial.print("0");}
Serial.print(second, DEC);
if(hour<12){
    Serial.print("am");}
else{
    Serial.print("pm");}
Serial.print(dayOfMonth,DEC);
delay(1000); }
Serial.print("-");
Serial.print(month,DEC);
Serial.print("-");
Serial.print(year,DEC);
Serial.println();
delay(1000);
}

```

Hasil eksekusi program pembacaan data DS1307 diatas diamati menggunakan serial monitor dengan setting *baud rate* 9600 bps. Hasil pengujian ditunjukkan oleh gambar 4.2.

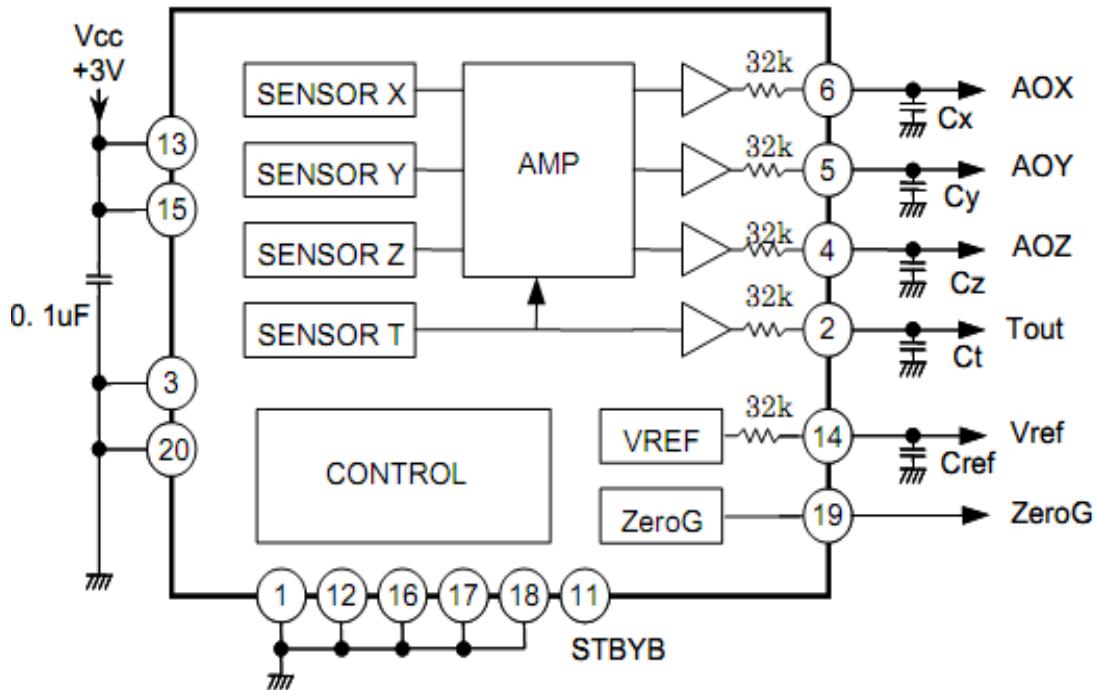


Gambar 4.2 Pembacaan data DS1307

IV.2 Metode Akusisi Data Percepatan

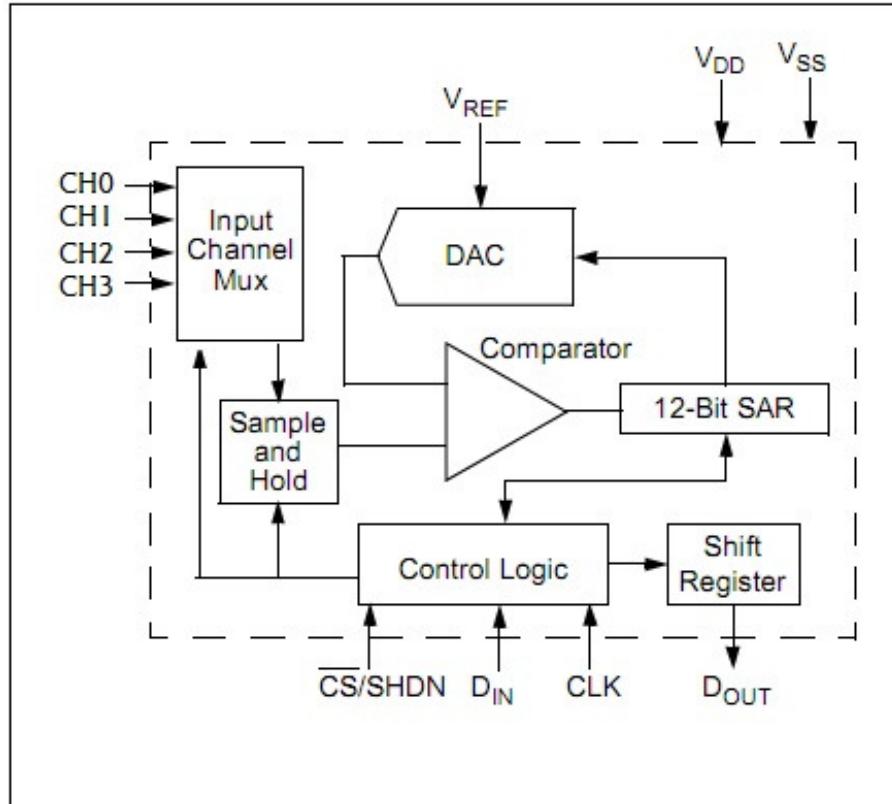
Dalam penelitian ini, pembacaan percepatan pergerakan tanah menggunakan modul *accelerometer* buatan Parallax.Inc dimana dalam modul ini sensor yang digunakan adalah H48C dan telah terintegrasi dengan *4 channel 12-bit A/D converter with serial interface MCP3204*.

Blok diagram dari H48C ditunjukkan oleh gambar 4.3. Dengan AOX sebagai analog output sumbu x, AOY sebagai analog output sumbu y, AOZ sebagai analog output sumbu z, dan Vref sebagai tegangan referensi. Maka, untuk mengetahui besarnya percepatan yang dinyatakan dalam level tegangan analog untuk tiap sumbu adalah sebagai berikut.



Gambar 4.3 Blok diagram H48C

Pada modul 3 axis accelerometer buatan Parallax.Inc output tegangan analog tiap sumbu dan output Vref dihubungkan dengan MCP3204 untuk mengkonversi data analog menjadi digital dan mengirimkanya ke kontroler secara serial.

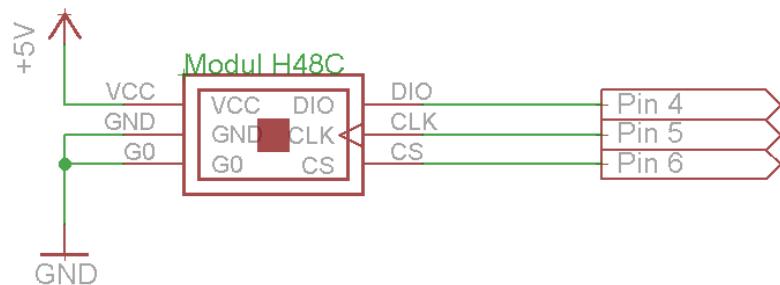


Gambar 4.4 Diagram Blok MCP3204

Diagram blok MCP3204 ditunjukkan oleh gambar 4.4. Pada modul accelerometer, pin AOX pada H48C dihubungkan ke CH0 dari MCP3204, AOY pada H48C terhubung dengan CH1 dari MCP3204, AOZ pada H48C terhubung dengan CH2 dari MCP3204 dan Vref pada H48C terhubung dengan CH3 dari MCP3204.

IV.2.1 Skematik Rangkaian Accelerometer

Rangkaian pengujian menggunakan modul 3 axis *accelerometer* H48C dan kontrolernya menggunakan *Arduino board* tipe Deumilanove. Skematik rangkaian pengujian ditunjukkan oleh gambar 4.5.



Gambar 4.5 Skematik rangkaian pengujian modul accelerometer

Dari modul accelerometer hanya dibutuhkan 3 pin untuk melakukan pembacaan, yakni :

1. Pin DIO sebagai jalur digital input / output, terhubung dengan pin 4 *Arduino board*.
 2. Pin CLK untuk pemberian *clock trigger* sensor dan terhubung dengan Pin 5 *Arduino board*.
 3. Pin CS (*Chip Select*) untuk aktifasi modul, terhubung dengan Pin 6 *Arduino board*.

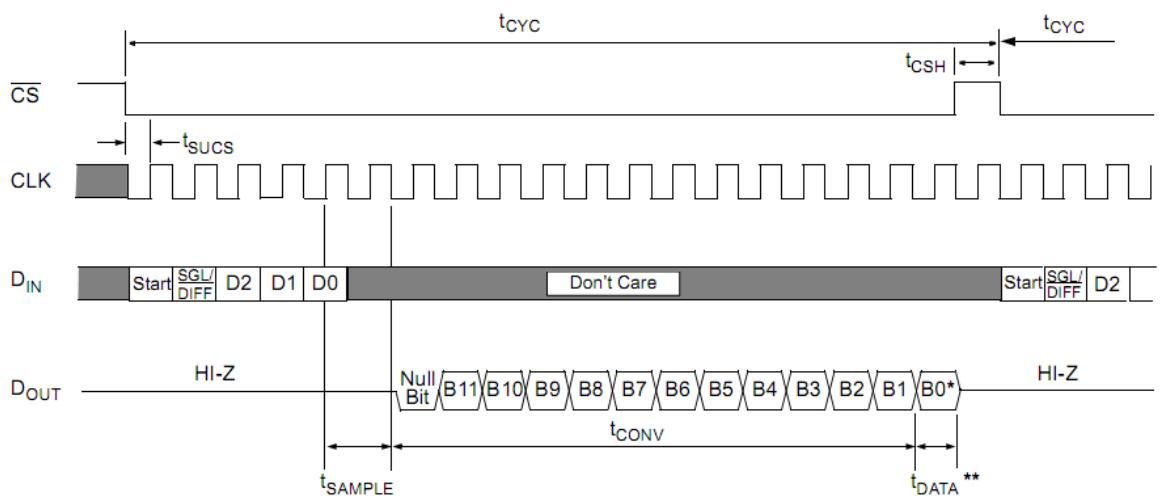
IV.2.2 Algoritma Pembacaan Percepatan Tiap Sumbu

Pembuatan algoritma pembacaan tiap sumbu didasarkan pada timing diagram dari MCP3204 (gambar 4.6). Frekuensi *clock* optimal dari MCP3204 adalah sebesar 10 KHz sehingga periode 1 *clock* yang paling optimal dapat dihitung dengan persamaan dibawah(persamaan 4.4).

Sehingga periode *clock* (t_{clock}) optimal adalah 0,0001 s atau sebesar 100 μ s dan dalam satu pulsa *clock* terdapat 1 periode pulsa

bernilai rendah (t_{low}) dan 1 periode pulsa bernilai tinggi (t_{high}) sehingga masing – masing t_{low} dan t_{high} memiliki selang waktu $t_{low/high} = \frac{t_{clock}}{2}$ yakni sebesar $50 \mu\text{s}$.

Dalam modul *accelerometer* yang digunakan, jalur D_{IN} dan D_{OUT} digabungkan menjadi 1 dan disebut pin DIO.



Gambar 4.6 Timing Diagram MCP3204

Karena pin D_{IN} dan D_{OUT} digabungkan menjadi 1 pin, maka untuk memungkinkan pembacaan data sensor diperlukan kontroler untuk mengatur kondisi pin DIO sebagai output pada saat pengiriman *control bit* dan sebagai input pada saat pembacaan data hasil konversi MCP3204.

Konfigurasi kontrol bit untuk pembacaan tiap *channel* sesuai dengan tabel 4.1.

Mengingat bahwa CH0 = AOX, CH1 = AOY, CH2 = AOZ, CH3 = Vref maka alternatif konfigurasi input yang paling mudah

adalah dengan input *single – ended* yakni dengan men-set bit Single/Diff bernilai 1.

Tabel 4.1. Konfigurasi Bit Kontrol MCP3204

Control Bit Selections				Input Configuration	Channel Selection
Single/ Diff	D2*	D1	D0		
1	X	0	0	single-ended	CH0
1	X	0	1	single-ended	CH1
1	X	1	0	single-ended	CH2
1	X	1	1	single-ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+

Notasi algoritmik untuk pembacaan modul *accelerometer* adalah sebagai berikut .

```

Procedure getH48C ( input=D1,D0)
    Set DIO sebagai Output
    Set CLK sebagai output
    Set /CS sebagai output
    {--Kirim start bit--}
    CLK ke 1 (Falling edge), CS=0, DIO=1
    {--Kirim kontrol bit--}
    CLK ke 2, DIO=1 {--single ended--}
    CLK ke 3, DIO=0 {--D2 (don't care)--}
    CLK ke 4, DIO=D1
    CLK ke 5, DIO=D0
    {--D1.D0 = 00 → sb x
     D1.D0 = 01 → sb y
     D1.D0 = 10 → sb z
     D1.D0 = 11 → sb Vref--}
    CLK ke 6 {--sample time--}
    CLK ke 7 {--nul bit--}
    Set DIO sebagai Input
    {--ambil data dan simpan di array--}

```

```

CLK ke 8, dValue[11] = DIO
CLK ke 9, dValue[10] = DIO
CLK ke 10, dValue[9] = DIO
CLK ke 11, dValue[8] = DIO
CLK ke 12, dValue[7] = DIO
CLK ke 13, dValue[5] = DIO
CLK ke 14, dValue[6] = DIO
CLK ke 15, dValue[4] = DIO
CLK ke 16, dValue[3] = DIO
CLK ke 17, dValue[2] = DIO
CLK ke 18, dValue[1] = DIO
CLK ke 19, dValue[0] = DIO
For i=11 to i=0 begin
    Data += dValue[i]*2^i
End.
CS=1
Return data
{--end procedure--}

```

Dari notasi algoritmik diatas, kemudian dikonversi kedalam bahasa C versi arduino karena kontroler yang digunakan adalah *arduino board*. Listing program procedure gatH48C(D1,D0) pada arduino IDE adalah sebagai berikut.

```

int getH48C(int D1,int D0){
    digitalWrite(Cs, LOW);
    pinMode(Dio, OUTPUT);
    //***1
    digitalWrite(Clk, LOW);
    digitalWrite(Dio, HIGH);// start bit
    delayMicroseconds(50);
    digitalWrite(Clk, HIGH);
    delayMicroseconds(50);
    //***2
    digitalWrite(Clk, LOW);
    digitalWrite(Clk, HIGH);
    delayMicroseconds(50);
    //***3
    digitalWrite(Clk, LOW);
    digitalWrite(Dio, LOW);// D2 = 0
    delayMicroseconds(1);
    digitalWrite(Clk, HIGH);
    delayMicroseconds(50);
    //***4
    digitalWrite(Clk, LOW);
    ...
}

```

```

if (D1==0){ digitalWrite(Dio, LOW);}
else{ digitalWrite(Dio, HIGH);}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***5
digitalWrite(Clk, LOW);
if (D0==0){ digitalWrite(Dio, LOW);}
else{ digitalWrite(Dio, HIGH);}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***6
digitalWrite(Clk, LOW);
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***7
pinMode(Dio, INPUT);
digitalWrite(Clk, LOW);
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***8
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[11]= 1 ;
}
else {
    dValue[11]= 0 ;
}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***9
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[10]= 1 ;
}
else {
    dValue[10]= 0 ;
}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***10
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[9]= 1 ;
}
else {
    dValue[9]= 0 ;
}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***11
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
dValue[8]= 1 ;
}
else {

```

```

        dValue[8]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***12
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[7]= 1 ;}
else {
    dValue[7]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***13
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[6]= 1 ;}
else {
    dValue[6]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***14
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[5]= 1 ;}
else {
    dValue[5]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***15
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[4]= 1 ;}
else {
    dValue[4]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***16
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[3]= 1 ;}
else {
    dValue[3]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***17
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[2]= 1 ;}
else {

```

```

        dValue[2]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***18
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[1]= 1 ;}
else {
    dValue[1]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
/***19
digitalWrite(Clk, LOW);
if (digitalRead(Dio)==HIGH){
    dValue[0]= 1 ;}
else {
    dValue[0]= 0 ;}
delayMicroseconds(50);
digitalWrite(Clk, HIGH);
delayMicroseconds(50);
data=(dValue[11]*2048)+(dValue[10]*1024)+(dValue[9]*512)+(dValue[8]*256)+(dValue[7]*128)+(dValue[6]*64)+(dValue[5]*32)+(dValue[4]*16)+(dValue[3]*8)+(dValue[2]*4)+(dValue[1]*2)+(dValue[0]*1);
digitalWrite(Cs,HIGH);
return data;
}

```

Untuk pembacaan nilai tegangan dilakukan dengan memanggil prosedur getH48C(D1,D0) dengan D1 = 1 dan D0 = 0 untuk membaca Vref, D1 = 0, D0 = 0 untuk membaca nilai sumbu X, D1 = 0, D0 = 1 untuk membaca nilai sumbu Y, dan D1 = 1 dan D0 = 0 untuk membaca nilai pada sumbu Z. Contoh pemanggilan prosedur untuk pembacaan nilai Vref, sumbu X, sumbu Y, dan sumbu Z adalah sebagai berikut.

```

int dX, dY, dZ = 0;
ref = getH48C(1,1);
dX = getH48C(0,0);
dY = getH48C(0,1);
dZ = getH48C(1,0);

```

Sesuai dengan persamaan 4.1, 4.2, dan 4.3 maka level tegangan untuk tiap sumbu dinyatakan dengan mengeksekusi perintah berikut.

```
float aX, aY, aZ = 0.00;  
aX = (((float)dX) - ((float)ref));  
aY = (((float)dY) - ((float)ref));  
aZ = (((float)dZ) - ((float)ref));
```

Nilai gX , gY , dan gZ masih dalam level tegangan yang dinyatakan dengan nilai ADC 12 bit (0 hingga 4095). Untuk menyatakan kedalam nilai g sesuai dengan jangkauan pembacaan sensor (-3 g hingga +3g) maka digunakan persamaan berikut.

$$g = \left(\frac{data\ Axis-Vref}{4095} \right) x \left(\frac{3.3}{0.3663} \right) \quad \dots \dots \dots \quad (4.5)$$

Berdasarkan persamaan 4.5, nilai 4095 adalah nilai maksimum ADC 12 bit, 3.3 adalah tegangan suplai H48C dan 0.3663 adalah nilai tegangan keluaran 1g. Persamaan diatas disederhanakan menjadi persamaan 4.6 berikut.

Sehingga, *source code* untuk pembacaan nilai g untuk tiap sumbu dengan satuan g adalah sebagai berikut.

```
float aX, aY, aZ = 0.00;
aX = (((float)dX) - ((float)ref)) *0.0022;
aY = (((float)dY) - ((float)ref)) *0.0022;
aZ = (((float)dZ) - ((float)ref)) *0.0022;
```

Hasil pengujian program untuk pembacaan nilai g menggunakan *source code* diatas dilihat menggunakan serial monitor dengan *baud rate* 9600 bps sesuai dengan gambar 4.7.

```

aX = -1.04   aY = 0.26   aZ = 9.43
aX = -0.91   aY = 0.22   aZ = 9.43
aX = -1.08   aY = 0.22   aZ = 9.43
aX = -1.08   aY = 0.24   aZ = 9.43
aX = -1.10   aY = 0.22   aZ = 9.41
aX = -1.10   aY = 0.22   aZ = 9.37
aX = -0.93   aY = 0.28   aZ = 9.39
aX = -1.01   aY = 0.28   aZ = 9.50
aX = -0.99   aY = 0.28   aZ = 9.39
aX = -0.91   aY = 0.26   aZ = 9.43
aX = -1.08   aY = 0.22   aZ = 9.43
aX = -1.10   aY = 0.22   aZ = 9.41
aX = -1.10   aY = 0.22   aZ = 9.37
aX = -1.10   aY = 0.19   aZ = 9.37
aX = -1.04   aY = 0.15   aZ = 9.37
aX = -0.97   aY = 0.28   aZ = 9.39
aX = -0.97   aY = 0.26   aZ = 9.43
aX = -0.97   aY = 0.30   aZ = 9.41
aX = -1.06   aY = 0.22   aZ = 9.41
aX = -0.95   aY = 0.19   aZ = 9.37
aX = -1.01   aY = 0.15   aZ = 9.50
aX = -1.04   aY = 0.19   aZ = 9.37
aX = -1.04   aY = 0.28   aZ = 9.34
aX = -1.06   aY = 0.26   aZ = 9.43

```

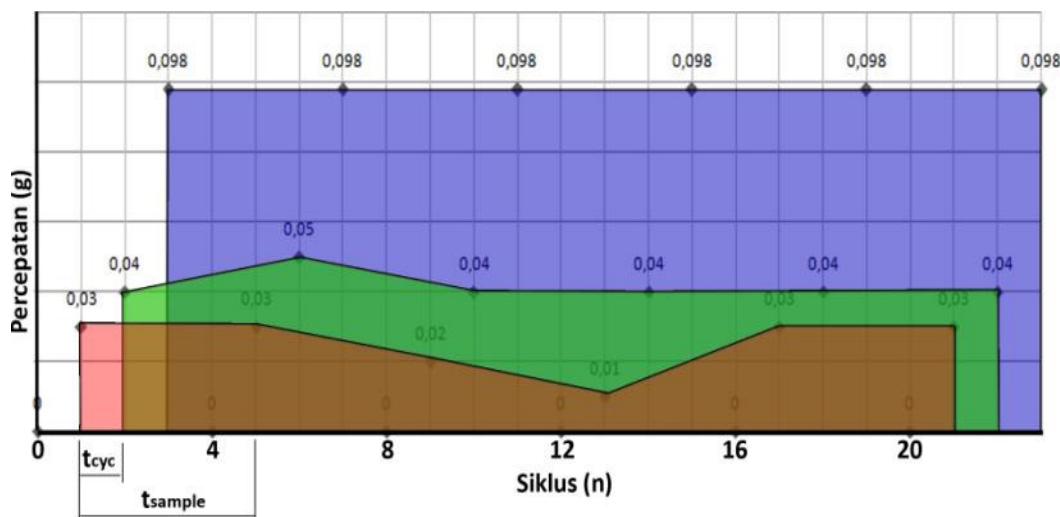
Gambar 4.7 Hasil pembacaan modul accelerometer

Pembacaan diatas memberikan hasil berupa percepatan sesaat tiap sampel. Dengan mengetahui t_{clock} dan jumlah $clock$ dalam satu siklus pengambilan data, maka dapat dihitung periode sampel dengan cara berikut.

- $t_{clock} = 100 \mu s$ dalam satu siklus pengambilan data terdapat 20 $clock$ sehingga periode 1 siklus (t_{cycle}) sebesar 20 kali $100 \mu s$ sehingga $t_{cycle} = 2000 \mu s$ atau 2 ms

- Algoritma pengambilan data dilakukan secara bergantian dari V_{ref} , dX , dY , dan dZ dan masing – masing proses pengambilan data merupakan 1 siklus pengambilan data sehingga periode sampel (t_{sample}) tiap sumbu sebesar 4 kali t_{cycle} sehingga $t_{sample} = 8 \text{ ms}$.

Sebagai ilustrasi ditunjukan oleh grafik dibawah (gambar 4.8) dimana untuk mengambil data sebanyak 6 sampel dari tiap sumbu diperlukan sebanyak 24 siklus pengambilan data. Siklus pertama ($n = 0$) diawali dengan pengambilan data V_{ref} , kemudian pada $n = 1$ diambil data percepatan sumbu x (daerah berwarna merah) kemudian $n = 2$ diambil data percepatan sumbu y (daerah berwarna hijau) kemudian dilanjutkan untuk $n = 3$ diambil data sumbu z (daerah berwarna biru) dan pada $n = 4$ diulangi lagi untuk pengambilan data V_{ref} dan seterusnya.



Gambat 4.8 Ilustrasi pengambilan data sebanyak 6 sampel untuk tiap – tiap sumbu

Setelah mengetahui periode sampel tiap sumbu adalah 8 ms maka jumlah sampel dalam 1 detik sebanyak 125 sampel.

IV.2.3 Karakteristik Akselerometer H48C

Akselerometer H48C dapat mengukur percepatan pada sumbu X, Y dan Z baik percepatan statik maupun percepatan dinamik.

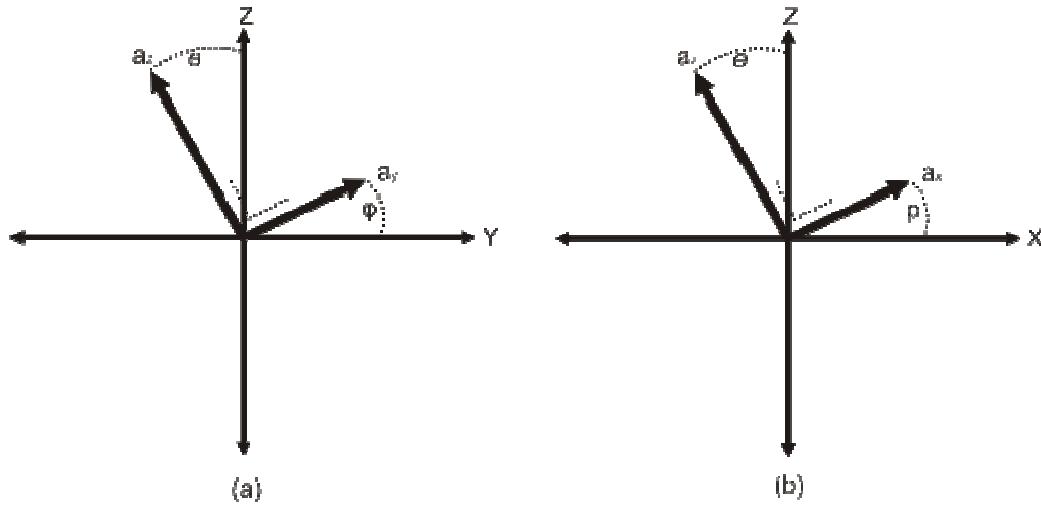
Percepatan statik merupakan percepatan yang dipengaruhi oleh gaya gravitasi. Sedangkan percepatan dinamik dipengaruhi oleh gaya dari luar selain gaya gravitasi.

a. Percepatan Statik

Pada kondisi pembacaan percepatan statik, nilai percepatan tiap sumbu dipengaruhi oleh kemiringan sensor terhadap bidang referensi dalam hal ini adalah gaya gravitasi. Sehingga kondisi statik ideal untuk aplikasi pembacaan kemiringan suatu objek.

Nilai percepatan yang diberikan merupakan representasi posisi dari sensor terhadap pusat gravitasi dan magnitude vektor percepatan dari 3 sumbu yang diberikan sama dengan 1g. Kemiringan sensor memiliki sudut *pitch* (ρ), *roll* (ϕ) dan *theta* (θ).

Pitch (ρ) merupakan sudut antara sumbu X relatif terhadap bumi. *Roll* (ϕ) merupakan sudut antara sumbu Y relatif terhadap bumi. Sedangkan *theta* (θ) merupakan sudut antara sumbu Y relatif terhadap gravitasi. Sudut ρ , ϕ , dan θ masing – masing ditunjukan oleh gambar 4.9 dalam representasi 2 dimensi.



Gambar 4.9. Sudut roll (ϕ) dalam plot dua dimensi antara sumbu Y dan Z (gambar a), Sudut pitch (ρ) dalam dua dimensi antara sumbu X dan Z.

Akselerometer mengukur percepatan statis pada tiap sumbu dan memberikan data berupa vektor percepatan sehingga kita dapat menghitung besar sudut yang terbentuk pada masing – masing sudut ρ , ϕ , dan θ dengan persamaan 4.8, 4.9 dan 4.10 berikut.

$$\theta = \arctan \left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z} \right) \dots \dots \dots \quad (4.10)$$

b. Pengujian Percepatan Statik

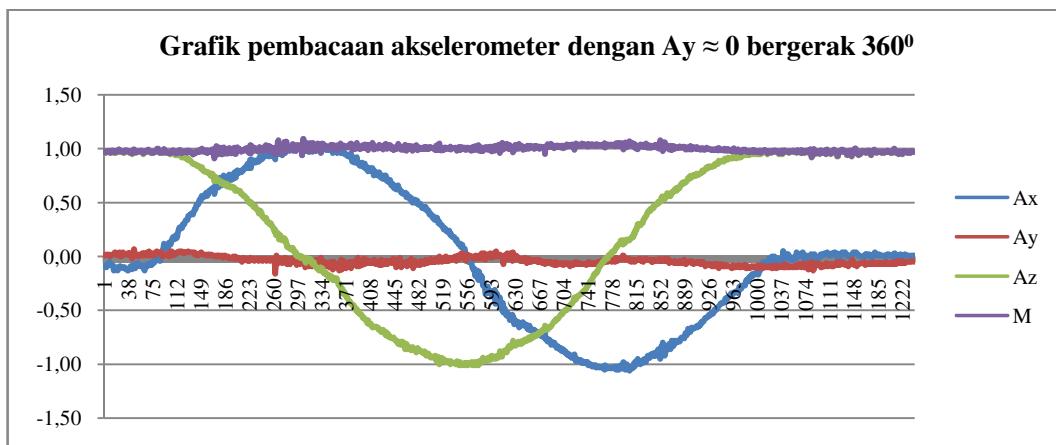
Pengujian pengukuran statik dilakukan untuk mengetahui karakteristik akcelerometer H48C pada saat bergerak statis.

Dalam pengujian ini dilakukan dua skema yaitu :

1. Pengambilan sudut ρ dan θ dengan cara memutar sensor pada sumbu X (sebesar satu putaran penuh (360^0)) dan kemudian memutar sensor sebesar $\frac{1}{4}$ putaran pada arah positif kemudian kembali lagi ketitik awal ($\frac{1}{4}$ putaran arah negatif).
2. Pengambilan sudut φ dan θ dengan cara memutar sensor pada sumbu Y sebesar satu putaran penuh (360^0)) dan kemudian memutar sensor sebesar $\frac{1}{4}$ putaran pada arah positif kemudian kembali lagi ketitik awal ($\frac{1}{4}$ putaran arah negatif)

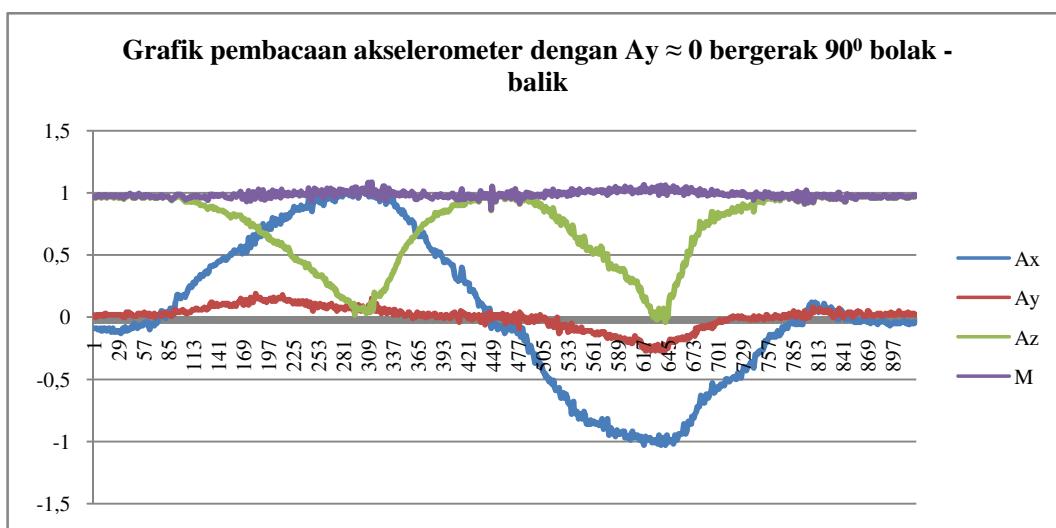
Data hasil pengujian ditunjukan pada lampiran 1. Nilai M (resultan) diisi dengan menggunakan persamaan $M = \sqrt{A_x^2 + A_y^2 + A_z^2}$. Nilai M diperlukan untuk menguji apakah sensor bekerja statis atau tidak. Dapat dilihat bahwa dari data percepatan yang diberikan (lampiran 1) akselerometer pada pengujian pertama, tampak bahwa $M \approx 1g$. Nilai resultan vektor percepatan pada kondisi statik tidak melebihi dari nilai percepatan gravitasi sedangkan adanya nilai yang lebih besar dari 1 dari percobaan pertama diatas dikarenakan sensor melakukan pembacaan pada saat sensor diputar yang artinya sensor menerima gaya dari luar yang nilainya kecil.

Apabila dari tabel diatas dibuat grafik maka vektor percepatan sumbu X (Ax) merupakan fungsi *sinus* dan vektor percepatan pada sumbu Z (Az) merupakan fungsi *cosinus*.



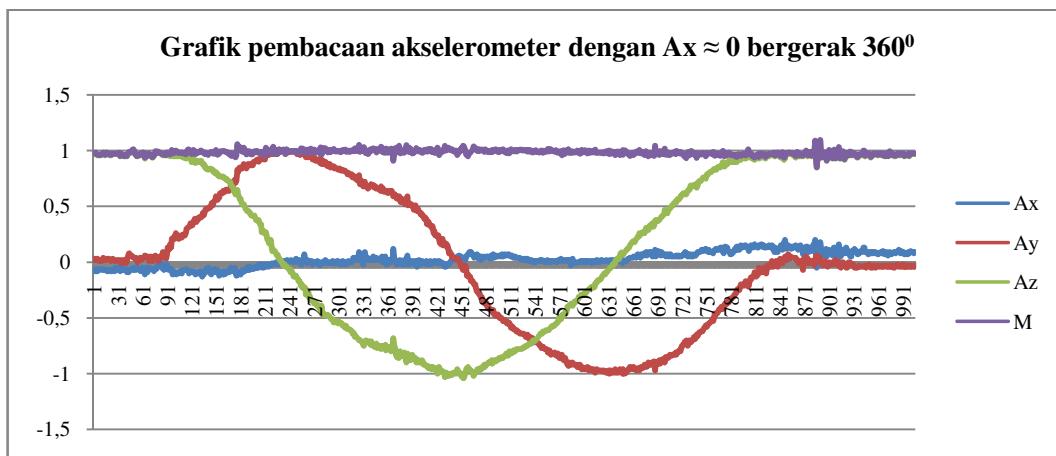
Gambar 4.10 Grafik percepatan pada akselerometer bergerak statik dengan $Ay \approx 0$ bergerak 360°

Pengujian kedua adalah menggerakan modul pada sumbu X (Sumbu Y = 0) sebesar 90° kearah positif kemudian kembali lagi sebesar 90° kearah negatif. Hasil pembacaan ditunjukan pada gambar 4.11 dibawah.

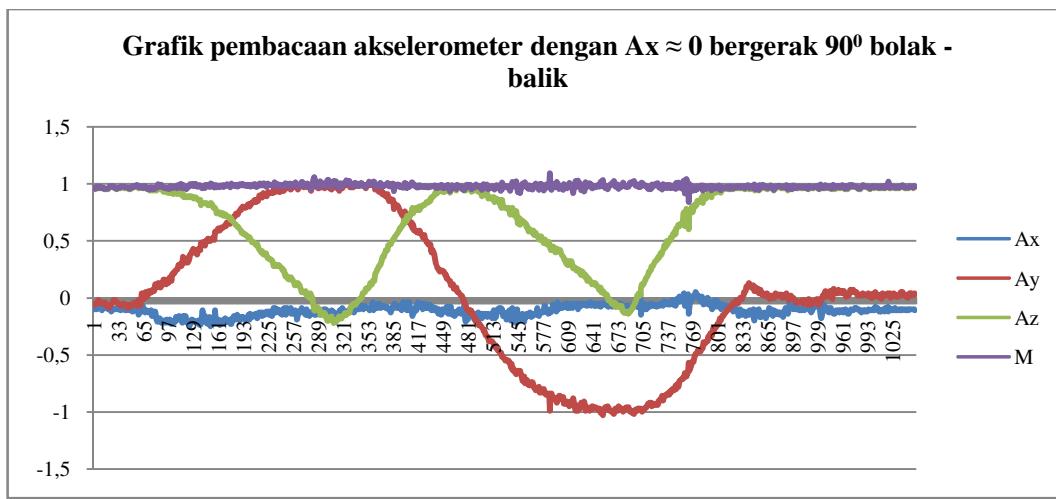


Gambar 4.11 Grafik percepatan pada akselerometer bergerak statik dengan $Ay \approx 0$ bergerak 90° bolak – balik.

Skema pengujian kondisi statik yang kedua adalah dengan memutar sensor pada sumbu Y ($X=0$) sebesar 360° dan dilanjutkan dengan memutar sensor pada sumbu Y sebesar 90° bolak balik. Didapat hasil sesuai gambar 4.12 dan 4.13.



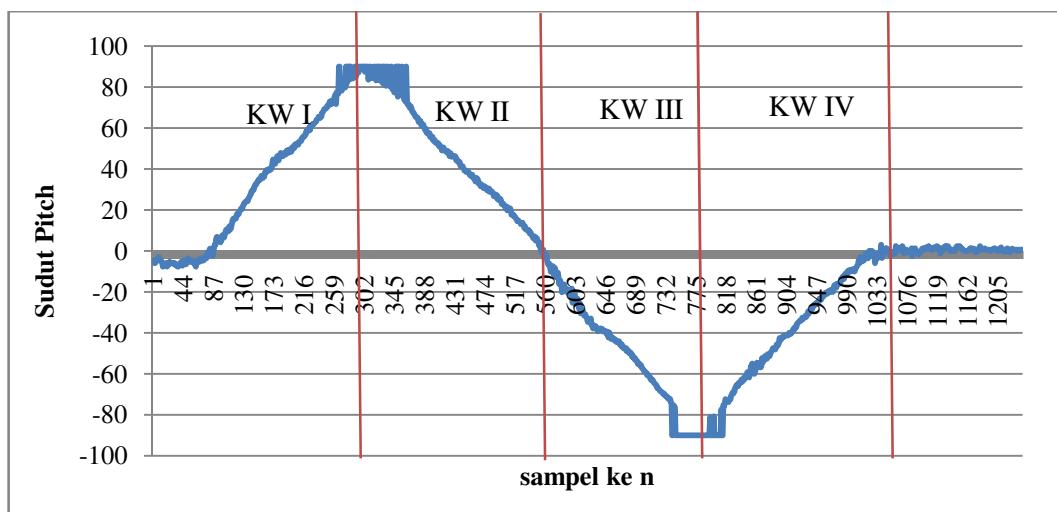
Gambar 4.12 Grafik percepatan pada akselerometer bergerak statik dengan $Ax \approx 0$ bergerak 360° .



Gambar 4.13 Grafik percepatan pada akselerometer bergerak statik dengan $Ax \approx 0$ bergerak 90° bolak – balik.

Baik vektor Ax maupun vektor Ay keduanya merupakan fungsi *sinus* sedangkan vektor Az merupakan fungsi *cosinus* sehingga sudut ρ, ϕ dan θ dapat dicari dengan menggunakan persamaan 4.8, 4.9 dan 4.10.

Akan tetapi fungsi *arctan* akan memberikan nilai pada jangkauan -90° hingga 90° sehingga persamaan 4.8, apabila diterapkan untuk menghitung sudut ρ dari hasil pembacaan akselerometer pada tabel 4.2 akan memberikan nilai dari -90° hingga 90° seperti pada gambar 4.14. Padahal pergerakan sesungguhnya adalah dari 0° hingga 360° . Fungsi *arctan* juga akan memberikan hasil yang sama untuk nilai sudut pada kuadran I dan kuadran III sebagai contoh untuk sudut 225° , nilai $\tan(225^\circ)=1$ sedangkan hasil dari fungsi *arctan* $1 = 45^\circ$. Persamaan 4.8, 4.9 dan 4.10 juga akan memberikan nilai yang salah apabila modul berputar pada kuadran II dan IV.



Gambar 4.14 Nilai yang didapat apabila menggunakan persamaan 4.8 untuk menghitung sudut ρ dari data akselerometer pada tabel 4.2.

Sudut – sudut pada kuadran II hingga kuadran IV dapat diperoleh dengan memodifikasi persamaan 4.8, 4.9, dan 4.10 sebagai berikut.

- Kuadran I

$$\rho = \arctan\left(\frac{Ax}{\sqrt{A_y^2 + A_z^2}}\right)$$

$$\varphi = \arctan\left(\frac{Ay}{\sqrt{A_x^2 + A_z^2}}\right)$$

$$\theta = \arctan\left(\frac{\sqrt{A_x^2 + A_y^2}}{Az}\right)$$

- Kuadran II

$$\rho = 180^0 - \arctan\left(\frac{Ax}{\sqrt{A_y^2 + A_z^2}}\right)$$

$$\varphi = 180^0 - \arctan\left(\frac{Ay}{\sqrt{A_x^2 + A_z^2}}\right)$$

$$\theta = 180^0 - \arctan\left(\frac{\sqrt{A_x^2 + A_y^2}}{Az}\right)$$

- Kuadran III

$$\rho = 180^0 - \arctan\left(\frac{Ax}{\sqrt{A_y^2 + A_z^2}}\right)$$

$$\varphi = 180^0 - \arctan\left(\frac{Ay}{\sqrt{A_x^2 + A_z^2}}\right)$$

$$\theta = 180^0 - \arctan\left(\frac{\sqrt{A_x^2 + A_y^2}}{Az}\right)$$

- Kuadran IV

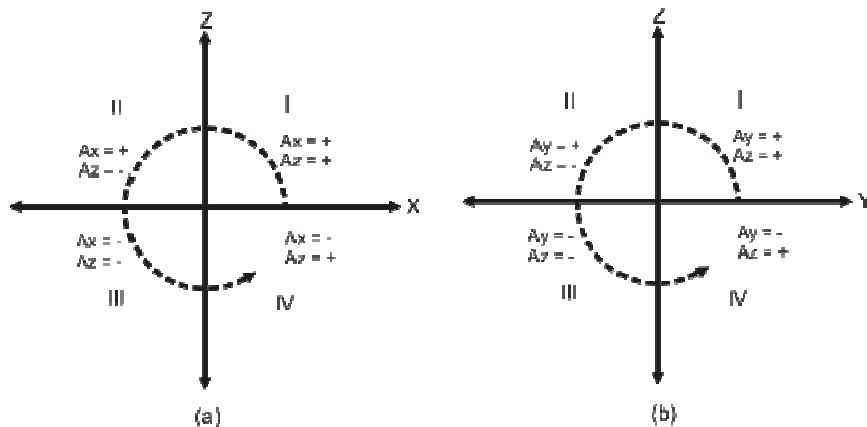
$$\rho = 360^0 + \arctan\left(\frac{Ax}{\sqrt{A_y^2 + A_z^2}}\right)$$

$$\varphi = 360^\circ + \arctan\left(\frac{Ay}{\sqrt{A_x^2 + A_z^2}}\right)$$

$$\theta = 360^\circ + \arctan\left(\frac{\sqrt{A_x^2 + A_y^2}}{Az}\right)$$

Untuk mengetahui diquadrant mana modul bergerak dapat dilakukan dengan membaca tanda bilangan positif (+) atau negatif (-) pada masing – masing data vektor percepatan.

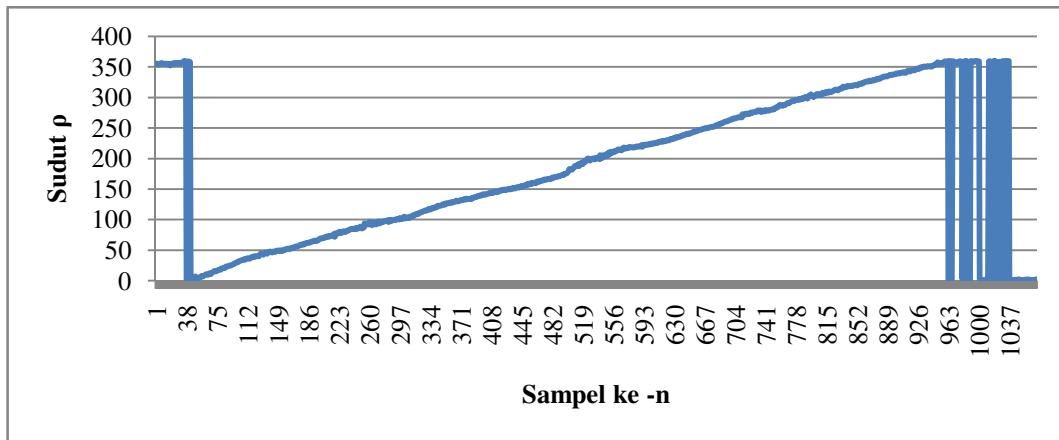
Penentuan daerah kuadran dapat dilakukan dengan analisis vektor percepatan pada tiap dua sumbu yaitu X-Z dan Y-Z. Berdasarkan analisa pada gerak melingkar 360° pada grafik yang ditunjukkan oleh gambar 4.10 dan 4.12 maka daerah kiadran I – IV dapat diketahui jika tanda bilangan vektor percepatan A_x , A_y dan A_z sesuai gambar 4.15.



Gambar 4.15 Penentuan daerah kuadran dengan menganalisa data vektor percepatan sumbu X dan Y terhadap sumbu Z.

Setelah mengetahui karakteristik perubahan nilai percepatan pada perpitaran dalam masing – masing kuadran langkah selanjutnya adalah mengaplikasikan persamaan yang didapat kedalam data

percepatan hasil pengujian. Hasil perhitungan menggunakan persamaan yang telah dimodifikasi untuk data pada tabel 4.2 ditunjukkan grafik pada gambar 4.16 dibawah. Sedangkan data selengkapnya mengenai pengujian percepatan statis dapat dilihat di lampiran 1.



Gambar 4.16 Nilai sudut ρ yang dihasilkan apabila pososisi kuadran diketahui

c. Percepatan Dinamik

Akselerometer H48C selain dapat mengukur keadaan percepatan statik juga dapat mengukur percepatan dinamik. Pada pengukuran percepatan dinamik, nilai percepatan yang didapat dipengaruhi oleh gaya gravitasi berikut gaya dari luar yang menyebabkan modul akselerometer bergerak.

Sehingga pada kondisi modul bergerak nilai resultant gaya dari akselerometer yang dibaca lebih besar dari 1g. Untuk mengetahui besar gaya yang bekerja pada modul yang menyebabkan modul bergerak dapat menggunakan persamaan berikut.

Dengan F_{total} merupakan jumlah keseluruhan gaya yang bekerja pada modul, F_g merupakan gaya gravitasi dan F_{eks} merupakan gaya eksternal yang bekerja pada modul. Dengan mengingat bahwa $F = m \cdot a$ maka persamaan 4.11 dapat dirubah menjadi.

$$m.a_{total} = m.a_g + m.a_{eks}$$

$$a_{total} = \frac{m \cdot a_g + m \cdot a_{eks}}{m}$$

Dimana, a_{total} adalah percepatan total yang bekerja pada benda, a_g adalah percepatan benda akibat pengaruh gaya gravitasi dan a_{eks} merupakan percepatan benda akibat gaya eksternal seperti dorongan atau tarikan.

Karena daerah kerja dari sensor merupakan bidang tiga dimensi maka persamaan 4.12 berlaku untuk masing – masing sumbu x, y dan z.

Sehingga:

$$a_x = a_g + a_{eks}$$

$$a_y = a_g + a_{eks}$$

$$a_z = a_g + a_{eks}$$

Ketika dalam kondisi diam vektor percepatan yang bekerja pada benda hanya karena pengaruh gaya gravitasi sedangkan ketika benda bergerak, vektor percepatan benda merupakan jumlah dari vektor percepatan akibat gaya gravitasi dan vektor percepatan akibat gaya eksternal. Sehingga percepatan akibat gaya eksternal dapat dicari

dengan mengurangi nilai percepatan pada saat benda bergerak dengan percepatan pada saat benda diam. Yang mana percepatan pada saat benda diam adalah a_g . Akan tetapi nilai percepatan eksternal tidaklah murni percepatan yang mengakibatkan benda bergerak, didalamnya terdapat vektor percepatan yang disebabkan oleh gaya gesekan benda, vibrasi, derau internal sensor dan lain sebagainya. Nilai percepatan ini dapat dikatakan sebagai faktor kesalahan (*error*). Sehingga percepatan benda bergerak sesungguhnya adalah sesuai persamaan 4.13 berikut.

d. Pengujian Percepatan Dinamik

Untuk pengujian percepatan dinamik dilakukan dengan mengambil data yang resultan vektor percepatanya lebih besar dari pada resultan vektor percepatan pada saat modul diam. Sehingga pada pengujian ini diperlukan data nilai percepatan pada saat modul tidak bergerak sebagai nilai acuan.

Data nilai percepatan awal pada saat modul diam diambil dari rata – rata nilai percepatan pada 50 sampel. Untuk proses inisialisasi ini diperlukan penambahan prosedur pada mikrokontroler dan menyimpan nilai resultan percepatan percepatan sebagai nilai acuan pengiriman data ke komputer.

Data yang dikirim ke komputer adalah data percepatan benda bergerak yang didalamnya terdapat faktor kesalahan atau *error* akan

tetapi untuk proses pengujian ini nilai *error* masih diikutsertakan untuk analisa lebih lanjut.

Prosedur inisialisasi data menggunakan *source code* sebagai berikut.

```
float x0 = 0;
float y0 = 0;
float z0 = 0;
float m0 = 0;
void inisialisasi(){
    for(int i=0;i<=50;i++){
        tX = getH48C(50,0,0);
        Xinit[i] = (((float)(tX-2048)))*0.0022;
        tY = getH48C(50,0,1) ;
        Yinit[i] = (((float)(tY-2048)))*0.0022;
        tZ = getH48C(50,1,0) ;
        Zinit[i] = (((float)(tZ-2048)))*0.0022;
        res[i] = sqrt(sq(Xinit[i])+sq(Yinit[i])+sq(Zinit[i]));
        m0 = max(m0 , res[i]);
        x0 = x0 + Xinit[i];
        y0 = y0 + Yinit[i];
        z0 = z0 + Zinit[i];
    }
    x0 = x0 * 0.02;
    y0 = y0 * 0.02;
    z0 = z0 * 0.02;
}
```

Procedure inisialisasi ini akan memberikan nilai pada variabel global x0,y0,z0 dan m0. Sedangkan data yang dikirim ke komputer merupakan data nilai percepatan yang resultan vektor percepatannya lebih besar dari m0 yakni dengan memanfaatkan percabangan *if*(....) sebagai berikut.

```
M=sqrt(sq(aX)+sq(aY)+sq(aZ));
if (M > m0){
    aX = aX - x0; //nilai percepatan dikurangi
    aY = aY - y0; //percepatan awal
    aZ = aZ - z0;
    Serial.print(aX);
    Serial.print(9, BYTE); //karakter Tab
    Serial.print(aY);
```

```
Serial.print(9,BYTE);
Serial.print(aZ);
Serial.print(9,BYTE);
Serial.println(M);
```

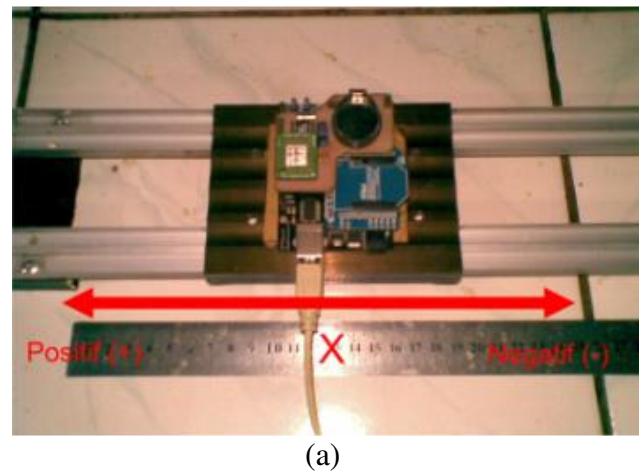
Kemudian setelah meng-*upload source code* ke mikrokontroler dilakukan percobaan menggrekan modul pada tiap sumbu sejauh jarak tertentu untuk mendapatkan karakteristik pergerakan modul. Tiap sumbu memiliki arah positif dan negatif sehingga pada uji coba dilakukan pengambilan data tiap sumbu sebanyak dua buah data yakni pada pegerakan positif dengan jarak tertentu dan pada pergerakan negatif pada jarak tertentu.

Pengujian untuk tiap sumbu di ilustrasikan oleh gambar 4.17. Untuk pengujian pergerakan tiap sumbu diambil data masing – masing sumbu untuk masing – masing arah pada perpindahan sejauh 10 cm, 20 cm, 30cm, 40cm serta 50 cm kemudian dibuat grafik untuk mempermudah analisa.

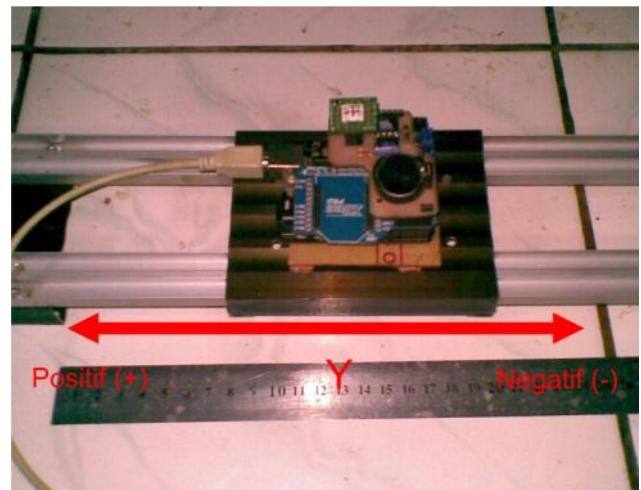
Data pengujian pergerakan pada sumbu x dengan arah positif ditunjukan pada tabel 4.2 , sedangkan grafik yang terbentuk ditunjukan oleh gambar 4.18.

Data pengujian pergerakan pada sumbu x dengan arah negatif ditunjukan pada tabel 4.3 , sedangkan grafik yang terbentuk ditunjukan oleh gambar 4.19.

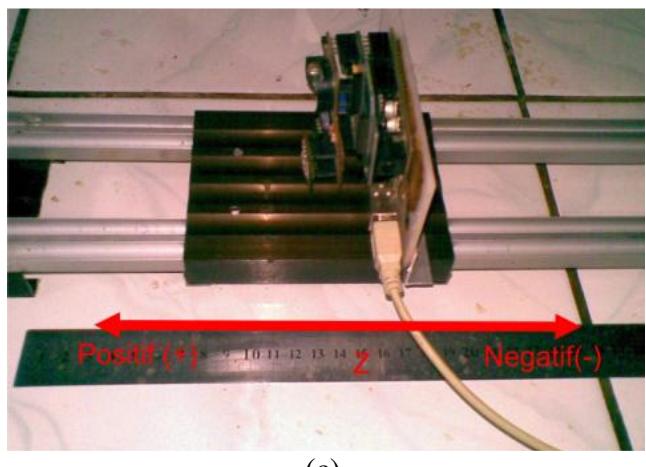
Data pengujian pergerakan pada sumbu y dengan arah positif ditunjukan pada tabel 4.4 , sedangkan grafik yang terbentuk ditunjukan oleh gambar 4.20.



(a)



(b)

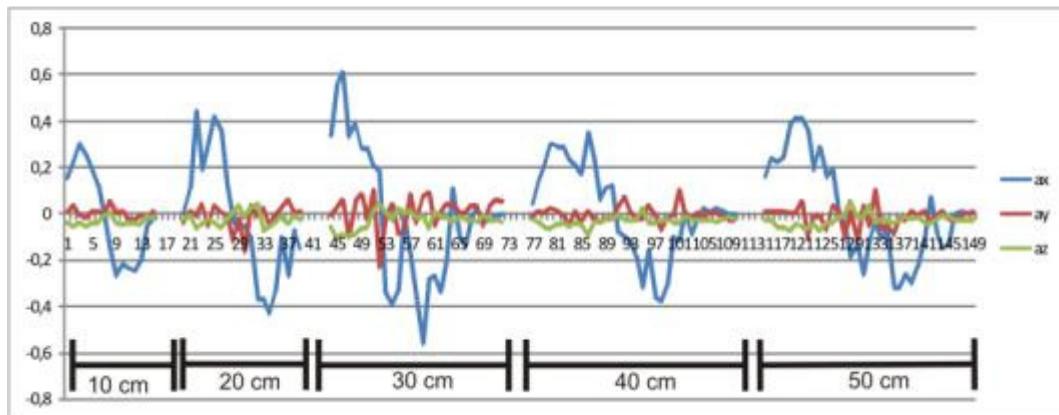


(c)

Gambar 4.17 Posisi modul pada saat pengujian pergerakan tiap sumbu. (a) sumbu x, (b) sumbu y dan (c) sumbu z

Tabel. 4.2 Data pengujian pergerakan pada sumbu x arah positif

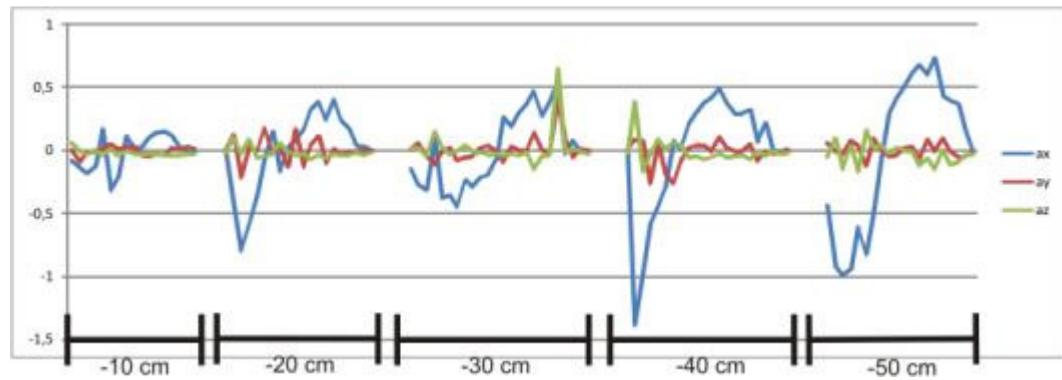
10 cm			20 cm			30 cm			40 cm			50 cm		
ax	ay	az												
0.15	0	-0	0	-0	-0	0.34	-0	-0.1	0.04	-0	-0	0.16	0.01	-0
0.22	0.03	-0.1	0.11	0.01	-0	0.55	0.02	-0.1	0.14	0.01	-0	0.24	0.01	-0
0.3	-0	-0	0.44	-0	-0.1	0.61	0.06	-0.1	0.21	0	-0.1	0.22	0.01	-0.1
0.25	-0	-0.1	0.19	0.04	-0.1	0.33	-0.1	-0.1	0.3	0.02	-0.1	0.24	0.01	-0.1
0.19	0.01	-0	0.3	-0.1	-0	0.39	0.05	-0.1	0.29	0.01	-0.1	0.38	0	-0.1
0.12	0.01	-0	0.42	0.03	-0	0.28	0.08	-0.1	0.29	-0	-0	0.41	0	-0.1
-0	-0	-0	0.36	0	-0.1	0.28	-0	-0.1	0.23	-0.1	-0.1	0.41	0.05	-0.1
-0.2	0.05	0	0.13	-0	-0	0.21	0.1	0.01	0.21	0.01	-0	0.36	-0.1	-0.1
-0.3	0	-0	-0	-0.1	0	0.18	-0.2	0.04	0.17	-0	-0.1	0.19	-0	-0.1
-0.2	0.01	-0.1	-0.1	-0	0.03	-0.3	-0	0.01	0.35	0.01	-0.1	0.29	-0	-0.1
-0.2	-0	-0	-0.1	-0.2	-0	-0.4	0.04	-0	0.23	-0	-0	0.16	-0.1	-0
-0.3	-0	-0	-0.1	0.03	0.02	-0.3	-0.1	0.02	0.06	-0	-0	0.19	0.03	-0
-0.2	-0	-0.1	-0.4	-0	0.04	-0	-0.1	0	0.11	-0	-0	0.01	0.01	-0
-0.1	-0	-0	-0.4	0.02	-0.1	-0.2	0.08	0.01	0.12	-0	-0	-0	-0.1	-0
-0	0.01	-0	-0.4	-0.1	-0.1	-0.4	-0	-0	-0.1	0.02	-0	-0.2	0.01	0.05
			-0.3	-0	-0	-0.6	0.07	-0	-0.1	0.07	-0	-0.1	-0.1	-0
			-0.1	0.02	-0	-0.3	0.09	-0.1	-0.1	0	-0	-0.3	0.03	-0
			-0.3	0.06	-0.1	-0.3	-0.1	0	-0.2	-0	-0	-0.1	-0.1	0.03
			-0.1	0	-0	-0.3	0	-0	-0.3	-0	0.02	-0	0.1	-0
			-0.2	0.01	-0	-0.2	0.04	-0	-0.2	0.03	-0	-0.1	-0.1	-0
						0.11	0.04	-0	-0.4	-0	-0	-0.1	-0.1	-0
						-0.1	0	-0	-0.4	-0.1	-0	-0.3	-0.1	-0.1
						-0.1	0	-0	-0.3	-0	-0	-0.3	-0	-0
						-0	0.03	-0	-0.1	-0	-0	-0.3	-0	-0
						0.02	0.03	-0	-0.1	0.1	-0	-0.3	0.01	-0
						-0	-0.1	-0	0.01	-0	-0	-0.2	-0	-0
						-0	0.03	-0	-0.1	-0	-0	-0.1	0.01	-0
						-0	0.06	-0	-0	0	-0	0.07	-0	-0
						0	0.05	-0	0.02	-0	-0	-0.1	-0	-0
								0	0.01	-0	-0.2	0.01	-0	
								0.02	0	-0	-0.1	-0	-0	
								0.01	-0	-0	0	0.01	-0	
								0	-0	-0	0.01	-0	-0	
								0	-0	-0	0	0	-0	
											0.01	0	-0	



Gambar 4.18 Grafik pengujian pergerakan modul pada sumbu x positif

Tabel 4.3 Data pengujian pergerakan sumbu x pada arah negatif

- 10 cm			- 20 cm			- 30 cm			- 40 cm			- 50 cm		
ax	ay	az												
-0.1	0	0.06	0	-0	-0	-0.2	0	0.02	0	0	-0	-0.4	0.06	-0.1
-0.1	-0.1	-0	-0.4	0.12	0.1	-0.3	0.06	0.02	-1.4	0.08	0.39	-0.9	-0	0.09
-0.2	-0	-0	-0.8	-0.2	-0	-0.3	-0.1	-0.1	-1	0.07	-0.2	-1	-0	-0.2
-0.1	-0	0	-0.6	0.03	0.08	0.09	-0.1	0.15	-0.6	-0.3	-0.1	-1	0.07	0.04
0.17	0.01	-0.1	-0.4	-0	-0.1	-0.4	-0	0.03	-0.5	0.06	0.08	-0.6	0.04	-0.2
-0.3	0.05	0.01	-0.1	0.18	-0.1	-0.4	0.02	-0	-0.3	-0.2	0.01	-0.8	-0.1	0.16
-0.2	0.01	-0.1	0.15	-0	-0	-0.5	-0.1	-0	0.07	-0.3	0.06	-0.5	0.09	0.03
0.11	0.03	-0	-0.2	0.02	0.06	-0.2	-0.1	0.04	0.01	-0.1	0.04	-0.1	0	0.04
0.03	0.03	-0	0.02	-0.1	-0	-0.3	-0.1	-0	0.2	0.02	-0.1	0.28	-0.1	-0
0.02	-0	-0	0.07	0.17	-0	-0.2	0.02	-0	0.29	0.04	-0.1	0.41	-0	0.02
0.09	-0.1	-0	0.16	-0.1	-0	-0.2	0.04	-0	0.37	0.04	-0.1	0.5	0.02	0
0.14	-0	-0	0.32	0.04	-0.1	-0.1	-0	-0	0.41	-0	-0.1	0.61	0.03	-0
0.15	-0.1	-0	0.38	0.11	-0	0.26	-0.1	-0	0.49	0.1	-0	0.67	-0.1	-0.1
0.11	0.02	-0	0.24	-0.1	-0.1	0.19	0.03	-0	0.37	0.02	-0.1	0.6	0.08	-0.1
0.03	0	-0	0.4	0.01	-0	0.29	0	-0.1	0.29	-0	-0.1	0.73	-0	-0.2
-0	0.03	-0	0.23	-0	-0	0.36	0	-0	0.29	-0	-0	0.43	0.09	0
-0	0.01	-0	0.17	-0	-0	0.47	0.14	-0.2	0.32	0.05	-0.1	0.39	-0	-0.1
			0.04	-0	-0	0.27	0	-0.1	0.06	-0.1	-0	0.36	-0.1	-0.1
			0.03	0	-0	0.37	-0	-0	0.21	-0	-0.1	0.15	-0.1	-0
			-0	0	-0	0.53	0.39	0.65	0	-0	-0	-0	-0	-0
						-0	0.11	-0	-0	-0	-0			
						0.07	-0.1	0.01	0	0.01	-0			

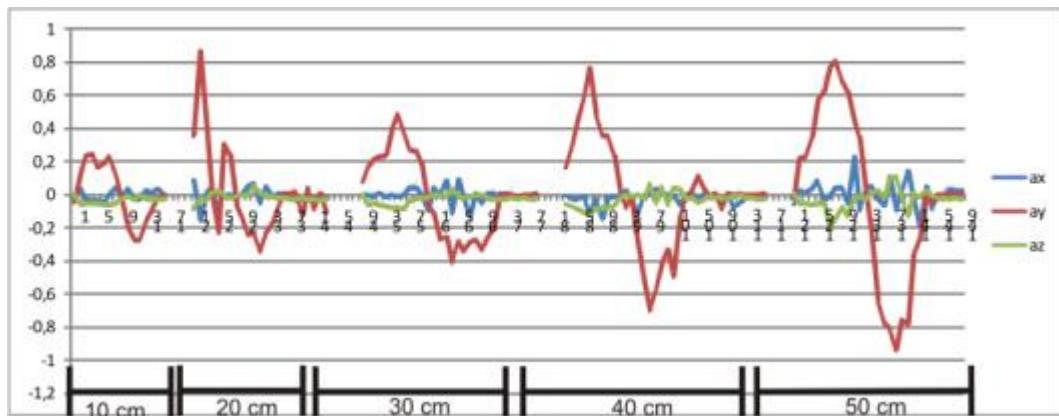


Gambar 4.19 Grafik pengujian pergerakan modul pada sumbu x negatif

Tabel 4.4 Data pengujian pergerakan sumbu y pada arah positif

10 cm			20 cm			30 cm			40 cm			50 cm		
ax	ay	az												
-0	-0	0	0.08	0.36	-0.1	0	0.08	0	-0	0.16	-0.1	-0.1	-0	0.01
0.03	0.14	-0.1	-0.2	0.87	-0	0	0.16	-0.1	-0	0.3	-0.1	0.02	0.22	-0.1
-0	0.24	-0.1	0.01	0.41	-0	-0	0.21	-0.1	-0	0.45	-0.1	0.01	0.23	-0.1
-0	0.25	-0.1	0.04	0.01	0.01	0.01	0.23	-0.1	-0	0.59	-0.1	0.03	0.35	-0.1
-0	0.16	-0.1	-0.1	-0.2	0.02	-0	0.24	-0.1	-0.1	0.77	-0.1	0.08	0.56	-0.1
-0.1	0.19	-0.1	0	0.31	0	0	0.38	-0.1	-0	0.49	-0.1	-0	0.62	-0.1
0	0.23	-0.1	0	0.24	-0	-0	0.49	-0.1	-0.2	0.36	-0.1	-0	0.76	-0.2
0.04	0.12	-0.1	-0	-0.1	-0	0	0.37	-0.1	-0	0.36	-0.1	0.04	0.81	-0.2
-0	-0.1	-0	0	-0.1	0	0.04	0.27	-0	-0	0.24	-0.1	0.04	0.68	-0.1
0.03	-0.2	-0	0.05	-0.3	0	0.04	0.26	-0	-0	0.04	-0	-0.1	0.62	-0.1
-0	-0.3	-0	0.07	-0.2	0.05	0	0.19	-0	0.02	-0.1	0	0.23	0.44	-0
-0	-0.3	0	-0.1	-0.3	0.01	-0.1	-0.1	-0	-0	-0	-0	-0.1	0.35	-0
0.02	-0.2	-0	0.05	-0.2	-0	0.04	-0.1	0	-0.1	-0.3	0	0.03	0.01	-0
0.01	-0.1	-0	-0	-0.2	-0	-0	-0.3	0	0	-0.5	-0	0.05	-0.2	0.01
0.03	-0	-0	0.01	-0.1	-0	0.08	-0.3	-0	0	-0.7	0.07	-0	-0.6	0.02
-0	0	-0	-0	0.01	-0	-0.1	-0.4	0.03	0.03	-0.6	-0.1	-0.1	-0.8	-0
			-0	0.01	-0	0.09	-0.3	-0	0.01	-0.4	0.05	0.11	-0.8	0.1
			-0	0.02	-0	-0	-0.3	-0	0	-0.3	-0.1	-0.1	-0.9	0.11
			-0	-0.1	-0	-0.1	-0.3	-0	0	-0.5	0.04	0.01	-0.8	0
			-0	0.04	-0	0.01	-0.3	0	-0.1	-0.1	0.03	0.14	-0.8	-0.1
			-0	-0.1	-0	-0.1	-0.3	0	-0	-0	-0	-0	-0.4	0.01
			0	0.01	-0	0.01	-0.3	-0	0.01	0.02	-0	-0.2	-0.3	-0
			-0	-0.1	-0	0.01	-0.2	-0	-0	0.11	-0.1	0.05	0.01	0.02

					-0	0.01	-0	0	0.04	-0	-0.1	-0.1	-0
					0.01	0	-0	-0	-0	-0	0	0	-0
					0	0	-0	-0	0.01	-0	-0	0	-0
					-0	-0	-0	-0	-0.1	-0	0.03	0	-0
					-0	0	-0	0	0.01	-0	0.02	0	-0
					0	0	-0	-0.1	0	-0	0.02	0.01	-0
					0.01	0	-0	-0.1	0.01	0			
									-0	0	-0		
									-0	0	-0		
									0	0	-0		
									-0	0.01	-0		



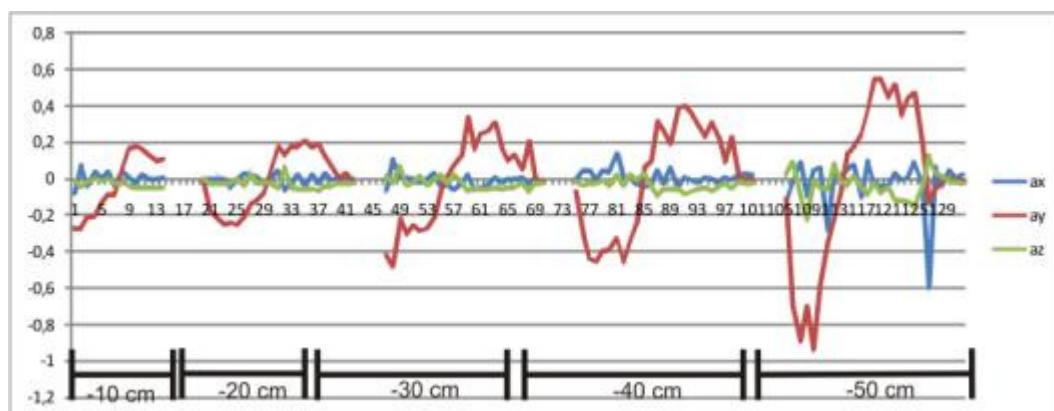
Gambar 4.20 Grafik pengujian pergerakan modul pada sumbu y positif

Data pengujian pergerakan pada sumbu y dengan arah negatif ditunjukkan pada tabel 4.5 , sedangkan grafik yang terbentuk ditunjukkan oleh gambar 4.21.

Tabel 4.6 Data pengujian pergerakan sumbu y pada arah negatif

-10 cm			-20 cm			-30 cm			-40 cm			-50 cm		
ax	ay	az	ax	ay	az	ax	ay	az	ax	ay	Az	ax	ay	az
-0.1	-0.3	-0	0	-0	0	-0.1	-0.4	0.01	-0	-0.1	-0	-0.1	-0.1	0.02
0.08	-0.3	-0.1	0	-0.2	-0	0.11	-0.5	-0	0.05	-0.3	-0	-0	-0.7	0.09
-0	-0.2	-0	0	-0.2	-0	0.01	-0.2	0.07	0.05	-0.4	-0	0.09	-0.9	-0.1
0.04	-0.2	0	0	-0.3	-0	0.01	-0.3	-0	-0	-0.5	-0	-0.1	-0.7	-0.2
0	-0.1	-0	-0.1	-0.2	-0	0	-0.3	-0	0.05	-0.4	-0	0.05	-0.9	-0

0.04	-0.1	0	-0	-0.3	0	-0	-0.3	0.01	0.04	-0.4	-0	0.06	-0.6	-0.1
-0	-0.1	-0	0.03	-0.2	-0	0	-0.3	-0	0.14	-0.3	0.02	-0.3	-0.4	-0.1
0.04	0.06	-0	0.02	-0.1	0.02	0.03	-0.2	-0	0	-0.5	-0	0.05	-0.2	0.08
0.01	0.17	-0	0.01	-0.1	-0	-0	-0.1	0.02	0.01	-0.3	0.02	0.01	-0.1	-0
-0	0.18	-0.1	-0	-0.1	-0	-0	0.01	-0	-0	-0.3	-0	0.05	0.13	-0
0.02	0.16	-0.1	0	0.04	-0	-0.1	0.07	0.02	-0.1	0.06	0.03	0.08	0.17	0.01
0	0.13	-0.1	0.05	0.18	-0.1	-0	0.12	-0	-0	0.09	-0	-0.1	0.24	-0.1
0	0.1	-0.1	-0.1	0.13	0.07	0.02	0.34	-0.1	0.05	0.32	-0.1	0.1	0.37	-0.1
0.01	0.11	-0.1	-0	0.18	-0.1	-0.1	0.16	-0.1	-0	0.26	-0.1	-0	0.55	-0
			0.02	0.18	-0.1	-0.1	0.25	-0.1	0.06	0.19	-0.1	-0	0.55	-0.1
			-0	0.21	-0.1	-0	0.26	-0.1	-0.1	0.39	-0.1	-0	0.45	-0.1
			0.02	0.17	-0.1	0.01	0.31	-0.1	0.01	0.4	-0.1	0.03	0.52	-0.1
			-0	0.19	-0.1	-0	0.17	-0.1	-0	0.36	-0.1	-0	0.35	-0.1
			0.03	0.12	-0	0	0.1	-0.1	-0	0.3	-0.1	0.01	0.44	-0.1
			-0	0.07	-0	0	0.13	-0.1	0.01	0.23	-0.1	0.09	0.47	-0.2
			0.01	0	-0	0.01	0.05	-0	0	0.31	-0.1	-0	0.19	-0.1
			-0	0.03	-0	-0	0.21	-0.1	-0	0.23	-0.1	-0.6	-0.1	0.13
			0	-0	-0	0.01	-0	-0	0.01	0.09	-0	0.07	-0.1	0.03
						-0	-0	-0	-0	0.23	-0.1	-0	-0	0.01
									0.02	-0	-0	0.05	0	-0
									0.03	0	-0	0	-0	-0
								0.02	0	-0	0.02	-0	-0	-0



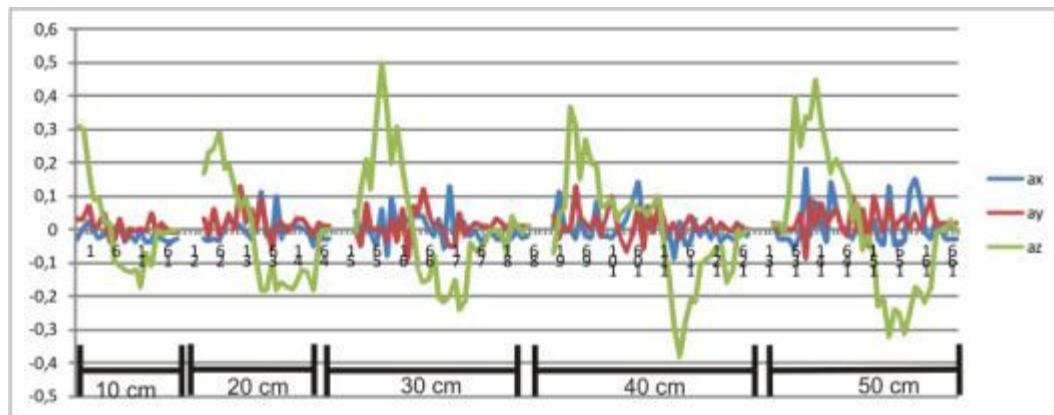
Gambar 4.21 Grafik pengujian pergerakan modul pada sumbu y negatif

Data pengujian pergerakan pada sumbu z dengan arah positif ditunjukan pada tabel 4.6 , sedangkan grafik yang terbentuk ditunjukan oleh gambar 4.22.

Tabel 4.6 Data pengujian pergerakan sumbu z pada arah positif

10 cm			20 cm			30 cm			40 cm			50 cm			
ax	ay	az													
-0	0.03	0.31	-0	0.03	0.17	0.05	0	-0	0.01	0.04	-0.1	0.01	0.02	0.01	
0	0.03	0.3	-0	-0	0.23	-0	-0.1	0.12	0.11	-0	0.04	-0	-0	0.02	
0.02	0.07	0.17	-0	0.06	0.24	0.03	0.08	0.21	-0	0.01	0.06	-0	0	-0	
0	0.01	0.09	-0	-0	0.29	-0	0	0.12	0	-0	0.37	-0	0	0.11	
-0	0.01	0.09	0	0.01	0.18	-0.1	0	0.32	-0	0.13	0.31	-0.1	0	0.4	
-0	0.05	0.02	0.03	0.05	0.19	0.06	0	0.5	0.04	0.04	0.15	-0	0.05	0.25	
0.01	-0	0.02	0.02	0	0.12	-0.1	-0	0.35	-0	0.02	0.27	0.18	-0.1	0.34	
0	-0	-0.1	0.01	0.13	0.06	0.09	0.03	0.2	-0	0	0.2	-0	0.09	0.33	
0.01	0.03	-0.1	-0	0.02	0.09	0.01	-0	0.31	0.08	0.04	0.19	0.08	0.01	0.45	
-0	-0	-0.1	-0	0.06	0.03	0.01	0.06	0.18	-0	-0	0.07	0.01	0.08	0.32	
-0	0	-0.1	0	-0	-0.1	0.02	-0.1	0.08	-0	0.03	0.07	-0	0.01	0.24	
-0	-0	-0.1	0.11	0.09	-0.2	0.06	0.07	-0	-0	0.1	0.09	0.14	0.03	0.17	
-0	0	-0.2	-0	0	-0.2	0.04	0.05	-0.1	-0	0.01	0.03	0.06	0.06	0.21	
-0	-0	-0.1	-0	-0.1	-0.1	0.04	0.12	-0.2	0	-0	0.05	0.01	0	0.18	
-0	0.05	-0.1	0.1	0.01	-0.2	0	0.05	-0.2	0.03	-0.1	0.06	-0	-0	0.14	
0	-0	0	-0	0.02	-0.2	-0	0.01	-0.1	0.08	-0	0.08	-0	0.1	0.03	
-0	0.02	-0	0.01	-0	-0.2	0.03	0.02	-0.2	0.14	0.07	0.06	0	-0	0.08	
-0	0	-0	0	0.01	-0.2	-0.1	0	-0.2	0	-0.1	0.06	0.05	0.06	-0.1	
-0	-0	0	0.01	0.03	-0.2	0.13	-0.1	-0.2	0.03	0.07	0.04	-0	-0.1	-0	
-0	-0	0	0	0.03	-0.1	-0	-0.1	-0.2	-0	-0	0.07	0.03	0.1	-0.1	
			-0	0.01	-0.1	0.02	0.05	-0.2	0.05	0.08	0.1	-0	0.05	-0.2	
			-0.1	-0	-0.2	0.02	-0	-0.2	-0	0.02	-0.1	-0.1	-0	-0.2	
			0.01	0.02	-0	-0	0	-0	-0	-0	-0.1	0.13	0.08	-0.3	
			-0	0.01	0	-0	0.02	-0.1	-0.1	0.02	-0.3	-0.1	0	-0.2	
			-0	0.01	0	-0	0.01	-0.1	0.02	-0	-0.4	-0	0.02	-0.3	
					-0	0.01	-0	0	-0	-0	0.04	-0.2	0.11	0	-0.3
					-0	0.03	0	0.03	-0	-0.2	0.15	0.05	0.05	-0.2	
					-0	0.02	-0	-0	0	-0.1	0.08	0	-0.2		
					-0	0.01	0	-0.1	0	0.01	-0.1	-0	0.03	-0.2	

					-0	0.02	0.04	-0	0.03	-0.1	-0	0.09	-0.2
					-0	0.01	0.01	0	-0	-0.1	0.03	0.01	-0
					-0	0.01	-0	-0	0.02	-0.1	-0	0.02	0
					-0	0.01	0.01	-0	0	-0.2	-0	0.02	0
								-0	-0	-0.1	-0	0.01	0.03
								-0	0.02	0	-0	0.02	0
								0.01	-0	-0			
								-0	0	0			



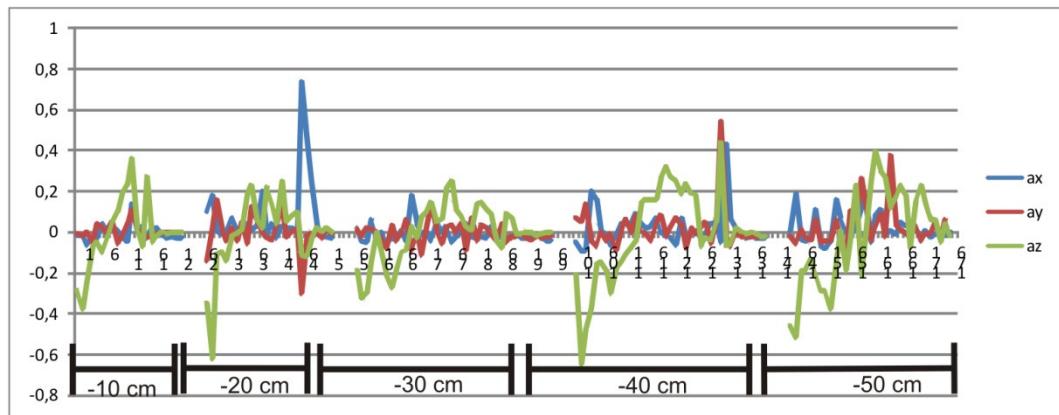
Gambar 4.22 Grafik pengujian pergerakan modul pada sumbu z positif

Data pengujian pergerakan pada sumbu z dengan arah negatif ditunjukkan pada tabel 4.7, sedangkan grafik yang terbentuk ditunjukkan oleh gambar 4.23.

Tabel 4.7 Data pengujian pergerakan sumbu z pada arah negatif

-10 cm			-20 cm			-30 cm			-40 cm			-50 cm		
ax	ay	az												
-0	-0	-0.3	0.1	-0.1	-0.4	0.02	0.01	-0.2	-0.1	0.07	-0.2	0.01	-0	-0.5
-0	-0	-0.4	0.18	0.02	-0.6	-0	-0	-0.3	-0.1	0.05	-0.7	0.19	-0.1	-0.5
-0.1	0	-0.2	0.01	0.16	-0.1	-0.1	0.02	-0.3	-0.1	0.14	-0.5	-0	0.01	-0.2
-0	-0.1	-0.1	-0	0.01	-0.1	0.06	0.01	-0.2	0.2	-0	-0.4	-0.1	-0	-0.2
-0	0.04	-0.1	0	-0	-0.1	-0	0	0	0.16	-0.1	-0.2	-0	-0	-0.1
0.04	0.02	-0.1	0.07	0.02	-0	0	-0	-0.1	0.01	0	-0.2	0.11	0.06	-0.2
0.01	0	-0	0.01	-0	-0	-0.1	-0.1	-0.2	-0	-0	-0.2	-0.1	-0.1	-0.3
0.03	0.05	0.04	0	0.05	0.01	-0	0.03	-0.3	-0.1	-0	-0.3	-0.1	-0	-0.3

0.01	-0.1	0.1	0.04	-0.1	0.18	-0	-0	-0.2	-0	-0.1	-0.2	-0.1	-0	-0.4
-0	-0	0.18	0.01	0.12	0.23	0.01	-0	-0.1	0.04	0.02	-0.2	0.16	0.06	-0.2
-0.1	0.03	0.23	0.03	0.06	0.06	-0	0.06	-0.1	0.04	0.06	-0.1	0.07	0	0.01
0.14	0.11	0.36	0.2	0	0.02	0.18	-0.1	0.03	0.04	0	-0.1	0	-0.1	-0.2
0.01	0.01	0.02	-0	-0	0.22	0.04	-0	-0	0.09	0.08	-0.1	-0	0.1	-0
0.03	-0	-0.1	0.04	-0	0.14	-0	-0.1	0.07	0.04	0	0.14	0.02	0.05	0.23
-0	-0	0.27	-0	0.02	0.04	0.01	0.03	0.1	0.02	-0	0.16	0.12	0.26	-0.2
0.02	-0	-0.1	0.03	0.13	0.25	-0	0.14	0.15	0.03	-0	0.16	0.18	0.14	-0
0.02	-0	-0	0.02	-0	0.05	0.05	0.01	0.05	0.07	0.04	0.16	-0.1	-0	0.25
-0	0	0	0.02	0	0.08	0	-0.1	0.06	0	0.08	0.26	0.08	0.02	0.4
-0	0	-0	0	0.01	0.1	0.03	0.02	0.21	-0	-0	0.32	0.11	0.09	0.3
-0	0	0	0.74	-0.3	-0.1	-0.1	0.03	0.25	-0	0.02	0.27	-0	-0	0.27
-0	0	0	0.45	-0	-0.1	-0	0	0.11	-0.1	0.07	0.25	0.01	0.37	0.12
-0	0	0	0.25	-0	-0	0	0.04	0.08	0.07	0.03	0.19	-0	0.03	0.17
			0.04	-0	0.02	0.04	-0.1	0.03	-0	-0.1	0.24	0.05	0.01	0.23
			-0	-0	0	-0	0.07	0.01	0	0.02	0.19	0.03	-0	0.18
			-0	0.01	0.02	-0	-0	0.14	0.01	-0	0.18	0.02	-0	-0.1
			-0	-0	0	-0	0.03	0.15	-0	0.02	-0.1	0.02	0.03	0.15
					-0	0.02	0.11	-0	0.05	-0	-0	-0	-0	0.23
					0.03	-0	0.09	0.04	-0.1	-0	0.01	-0	-0	0.14
					-0	-0	-0	0.05	0.06	0.05	-0	-0	0.06	
					-0	0.04	-0.1	-0.1	0.54	0.44	-0	0.04	0.06	
					-0	-0.1	0.09	0.43	0	-0.1	0	-0	-0.1	
					-0	-0	0.07	0.07	-0.1	-0.1	-0	0.06	0.04	
					-0	-0	-0	0.02	0	0.02	0	-0	-0	
					-0	-0	-0	-0	-0	0				
					-0	-0	0	-0	-0	-0				
					-0	-0	0	-0	-0	-0				
					-0	-0	-0	-0	-0	-0				
					-0	-0	0	-0	-0	-0				
					-0	-0	0	-0	-0	-0				

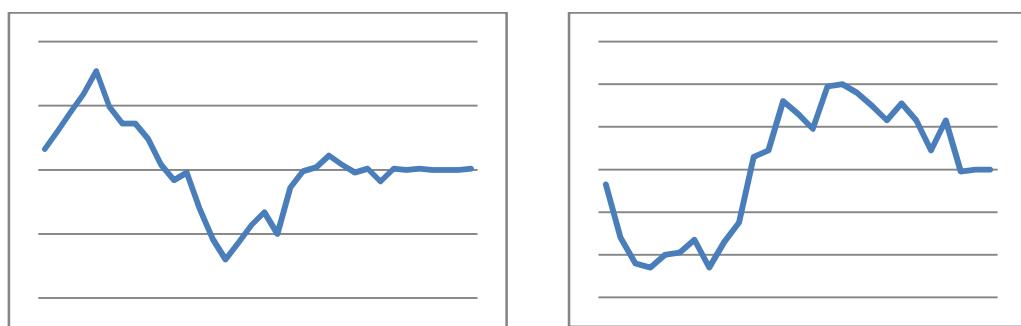


Gambar 4.23 Grafik pengujian pergerakan modul pada sumbu z negatif

e. Analisa Hasil Pengujian Percepatan Dinamik

1. Pergeseran ke arah negatif dan arah positif

Dapat diamati berdasarkan grafik yang diperoleh dari pengujian percepatan dinamik pada tiga sumbu diperoleh perbedaan bentuk grafik pada pergeseran pada arah positif dan arah negatif adalah sebagai berikut (gambar 4.24).



Gambar 4.24 Perbedaan arah pergerakan benda pada arah positif (kiri) dan arah negatif (kanan)

Gambar 4.24 merupakan contoh grafik yang terbentuk dari data percepatan pada perpindahan data sumbu y sejauh 40 cm pada arah positif dan 40 cm arah negatif. Pada pergerakan kearah positif, grafik

percepatan yang terbentuk seperti halnya gelombang sinus sedangkan pada arah pergerakan negatif grafik yang terbentuk seperti halnya gelombang sinus yang mengalami pergeseran fasa 180° .

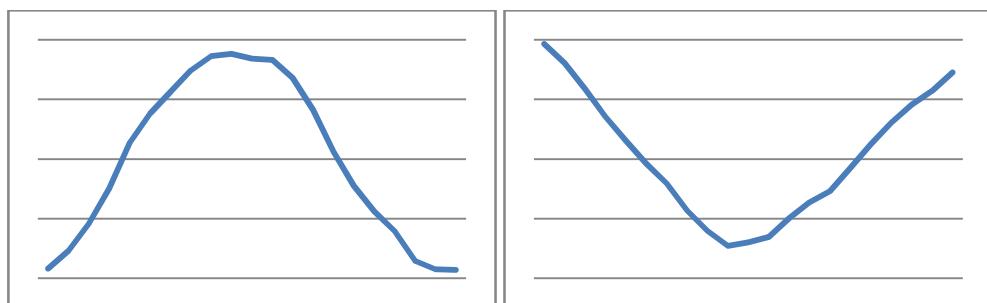
Dari grafik tersebut apabila di integrasikan akan didapat nilai kecepatan yang berbeda, yakni untuk pergerakan positif nilai kecepatan yang didapat bernilai positif sedangkan untuk pergerakan kearah negatif kecepatan yang didapat bernilai negatif. Integrasi data percepatan menjadi data kecepatan digunakan persamaan 2.4 yaitu

$$x_n = x_{n-1} + (y_n \cdot \Delta t)$$

Dengan percepatan ke-n adalah a_n dan kecepatan ke-n adalah v_n sehingga persamaan 2.4 diatas menjadi.

$$v_n = v_{n-1} + (a_n \cdot \Delta t)$$

Sedangkan nilai Δt adalah 8 ms sesuai dengan interval pengambilan data tiap sumbu sedangkan nilai v_0 adalah 0. Grafik pada gambar 4.24 diatas jika diintegrasikan menjadi seperti ditunjukan oleh gambar 4.25.

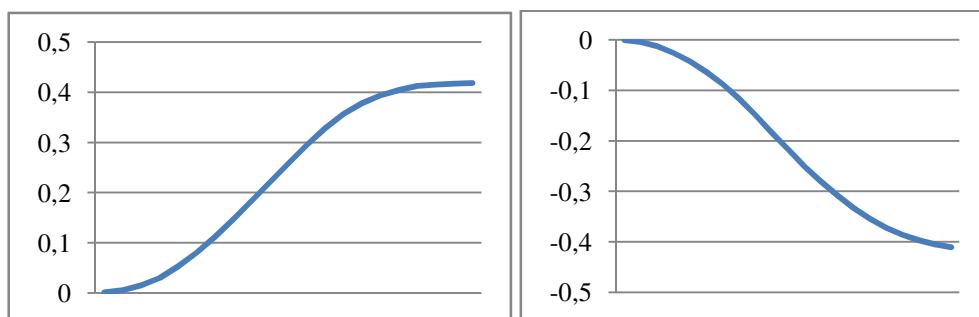


Gambar 4.25 Hasil integrasi data percepatan pada pergerakan benda pada arah positif (kiri) dan arah negatif (kanan)

Integrasi kedua akan menghasilkan data jarak perpindahan, parameter yang di integrasi adalah data percepatan sehingga persamaan yang digunakan adalah sebagai berikut.

$$s_n = s_{n-1} + (v_n \cdot \Delta t)$$

Untuk perhitungan ini nilai Δt masih sama dengan yang sebelumnya yaitu 8 ms sedangkan inisialisasi jarak perpindahan pada sample ke-0 adalah sejauh 0 m sehingga $s_0=0$. Hasil integrasi ditunjukkan oleh gambar 4.26 dengan total jarak perpindahan yang terjadi pada benda adalah pada sampel terakhir.



Gambar 4.26 Hasil integrasi data kecepatan pada pergerakan benda pada arah positif (kiri) dan arah negatif (kanan)

Dari gambar diatas dapat dilihat bahwa perpindahan benda hasil perhitungan mendekati perpindahan benda sesungguhnya serta dapat disimpulkan bahwa sensor mampu mendeteksi pergerakan linear pada arah positif maupun negatif dengan melihat nilai jarak perpindahan yang didapat yakni pada pergerakan kearah positif jarak perpindahan yang didapat bernilai positif sedangkan pada pergerakan kearah negatif jarak perpindahan yang didapat bernilai negatif.

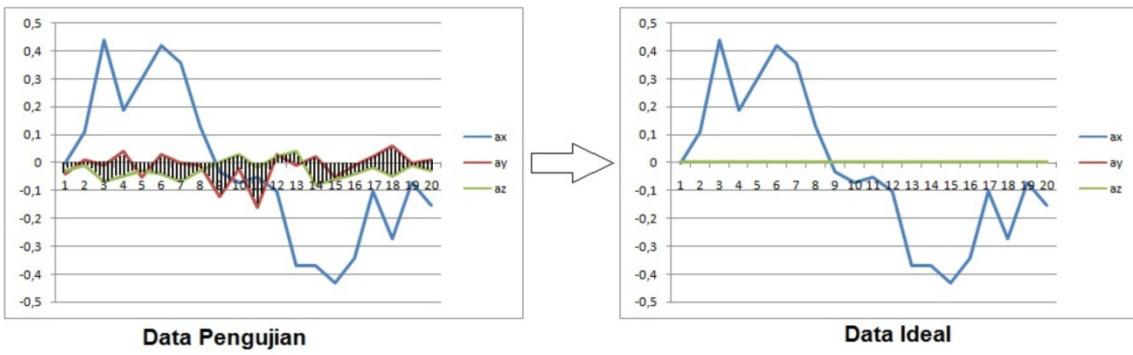
2. Pengaruh Pergerakan Satu Sumbu Terhadap Sumbu Yang Lain

Idealnya, jika akselerometer digerakan hanya pada satu sumbu secara linear maka nilai percepatan pada sumbu yang lain bernilai tetap karena tidak mengalami perubahan posisi. Misalnya jika sensor digerakan pada sumbu x kearah positif, maka nilai percepatan yang berubah seharusnya hanya percepatan pada sumbu x saja sedangkan nilai percepatan pada sumbu y dan z harusnya tetap.

Akan tetapi pada pengujian ini ternyata memberikan hasil yang berbeda karena meskipun digerakan pada satu sumbu tetapi nilai percepatan sumbu yang lain ternyata ikut berubah. Hal ini terjadi karena beberapa faktor diantaranya:

- a. Efek vibrasi pada saat modul bergerak.
- b. Gaya gesek media yang digunakan pada pengujian.
- c. Efek dari *Micro Electro-Mechanical System* (MEMS) yang diadopsi oleh akselerometer H48C.
- d. Pengaruh *noise* internal akselerometer.

Gambar 4.27 menunjukan data percepatan yang didapat pada pengujian dan perbandinganya dengan data ideal yang seharusnya didapat. Daerah yang diarsir adalah nilai percepatan pada sumbu y dan z yang ikut berubah ketika modul digerakan pada sumbu x saja.



Gambar 4.27 Perbandingan hasil pengujian dengan data ideal, meskipun digerakkan hanya pada sumbu x akan tetapi nilai percepatan pada sumbu y dan sumbu z juga ikut berubah.

Efek vibrasi dan pengaruh gaya gesek media pengujian merupakan kesalahan yang terjadi pada saat proses pengujian. Kesalahan ini merupakan faktor eksternal termasuk didalamnya adalah *human error*. Untuk meminimalisasi kesalahan ini diperlukan media pengujian yang benar – benar bagus diantaranya adalah media bersifat licin sehingga meminimalisasi gaya gesek antara media yang diam dengan media yang bergerak. Selain itu pada proses menggerakan benda juga diusahakan benda benar – benar terbebas dari dorongan maupun tekanan pada sumbu yang lain.

Efek dari *Micro Electro-Mechanical System* (MEMS) dapat dianalogikan dengan sistem pegas pada gambar 4.28. Sebuah kubus dengan masa M terhubung melalui dua buah pegas pada kedua sumbunya terhadap dinding sebuah balok. Pada kondisi diam, panjang regangan pegas pada masing – masing sumbu adalah x dan y . Sedangkan apabila balok diberi gaya dorong salah satu sumbu, misalnya sumbu x maka pada kubus M berlaku hukum *Newton I* yaitu

‘setiap benda memiliki sifat kelembaman’ yang mengakibatkan terjadinya perubahan regangan pada pegas x sejauh dx , akan tetapi pada pegas y juga mengalami perubahan panjang sejauh dy karena terjadi simpangan pada kubus M. Fenomena ini sama halnya jika sistem dalam representasi 3 dimensi. Sedangkan dalam teknik pengukuran inersial, fenomena ini disebut *cross axis effect*.



Gambar 4.28 Analogi Micro Electro-Mechanical System yang diadopsi oleh akselerometer H48C

3. Tingkat Akurasi Pembacaan Akselerometer

Tabel 4.8 menunjukkan data hasil pengujian akselerometer yang digerakkan pada sumbu x, y, dan z berikut selisih pembacaan dengan data sebenarnya.

Dari hasil yang ditunjukan pada tabel 4.8 menunjukan bahwa tingkat akurasi pembacaan akselerometer H48C tergolong cukup tinggi karena untuk keseluruhan pengujian rata – rata error dibawah 1.5 cm. Akan tetapi data satu sumbu diatas sebenarnya didalamnya masih terdapat noise yang disebutkan pada poin 2 diatas yang seharusnya tidak diikutsertakan dalam perhitungan. Hal ini dapat dibuktikan dengan melakukan perhitungan yang sama untuk sumbu yang lain yang seharusnya nilainya mendekati 0. Hasil perhitungan ditunjukan oleh

tabel 4.9 dibawah. Error rata – rata pengukuran ditunjukan dalam bentuk rMSE (*root Mean Square Error*).

Tabel 4.8 Data hasil pengujian pergerakan akselerometer pada sumbu x, y, dan z

Jarak (cm)	X			Y			Z		
	Terukur	Error	Akurasi	Terukur	Error	Akurasi	Terukur	Error	Akurasi
10	7.7	-2.3	77.0%	9.2	-0.8	92.0%	9.4	-0.6	94.0%
20	21.4	1.4	93.0%	21.2	1.2	94.0%	19.5	-0.5	97.5%
30	28.5	-1.5	95.0%	30.7	0.7	97.7%	29.8	-0.2	99.3%
40	41.8	1.8	95.5%	39.7	-0.3	99.3%	40.6	0.6	98.5%
50	51.5	1.5	97.0%	51.7	1.7	96.6%	52.9	2.9	94.2%
-10	-8	-2	80.0%	-10.7	0.7	93.0%	-9.5	-0.5	95.0%
-20	-19.6	-0.4	98.0%	-20.1	0.1	99.5%	-22.3	2.3	88.5%
-30	-30.2	0.2	99.3%	-32.3	2.3	92.3%	-31.8	1.8	94.0%
-40	-39.4	-0.6	98.5%	-40.4	0.4	99.0%	-38.1	-1.9	95.3%
-50	-51.1	1.1	97.8%	-53.9	3.9	92.2%	-53.4	3.4	93.2%
rMSE		1.44			1.63			1.83	

Tabel 4.9 Hasil *double* integrasi data percepatan pada sumbu yang tidak bergerak.

Jarak (cm)	Pergeseran sumbu x		Pergeseran sumbu y		Pergeseran sumbu z	
	y	z	x	z	x	y
10	0.3	4.6	-1.6	-5.2	-2.6	2.3
20	2.7	6.4	-1.9	-4.1	5.7	5.8
30	0.4	12.1	0	2.3	3.8	1.7
40	4.4	12.3	-2.5	-15.5	6	11.7
50	-4.5	-14.7	9	-9	0	1.9
-10	-1	-2	0	-2.5	-1.6	-0.2
-20	4.8	-4.1	2.4	0.4	-4.7	2.5
-30	-1.9	1.7	-1.8	-5.2	-0.9	-2.4
-40	-7.3	1.1	0.4	-7.3	-1.4	-2.8
-50	1	-4.1	1.4	-11	13	8.7

Bentuk grafik dari data percepatan, kecepatan dan jarak perpindahan dari hasil pengujian pergerakan satu sumbu pada akselerometer H48C dapat dilihat pada lampiran 2.

IV.2.4 Reduksi Error Pada Pengukuran Dan Perhitungan

Jika hanya menggunakan metode pengukuran dan perhitungan diatas, dapat diamati bahwa tingkat kesalahan dari pengukuran dan perhitungan cukup tinggi jika dibandingkan dengan nilai sesungguhnya.

Untuk meminimalisir tingkat kesalahan dilakukan perbaikan proses pengolahan sinyal lebih lanjut. Dimulai dari sinyal yang didapat dari sensor akselerometer.

Sinyal dari akselerometer terdapat banyak derau untuk itu diperlukan filter. Terdapat banyak macam filter digital yang dapat diaplikasikan untuk sinyal dari akselerometer ini. Pemilihan filter didasarkan pada sinyal yang didapat.

Data percepatan dari akselerometer merupakan sinyal yang informasinya direpresentasikan dalam runtun waktu diskret. Sehingga dipilih filter IIR (*Infinite Impulse Response*) dan *prototype* filter analog yang dipilih adalah LPF *Bessel Filter*. Fungsi rekursi untuk IIR ditunjukkan oleh persamaan 4.14.

$$y(n) = \sum_{k=0}^n b_k x(n-k) - \sum_{k=1}^n a_k y(n-k) \quad \dots \dots \dots \quad (4.14)$$

Langkah selanjutnya adalah menentukan parameter – parameter filter yaitu frekuensi sampling, frekuensi *cutt off*, dan orde filter.

Pengambilan data akselerometer dalam 1 detik sebanyak 125 data sehingga frekuensi sampling adalah 125 Hz, sedangkan frekuensi *cut off* dibuat 5 Hz.

Respon frekuensi dari filter digital adalah pada [0,1] sedangkan 1 merepresentasikan π yang merupakan frekuensi Nyquist sehingga $\pi = 62.5$ Hz. Frekuensi *cut off* dari filter digital adalah $\omega_c = f_c / f_s = 0.04$ rad/s.

Untuk mencari koefisien rekursi dari fungsi alih filter yang akan dibuat pada pengujian ini dilakukan dengan menggunakan program *Filter helper* yang tersedia pada website <http://www-users.cs.york.ac.uk/~fisher/mkfilter> dengan memasukan parameter frekuensi sampling, frekuensi *cut off* (*corner frequency*) dan orde filter.

a. Penentuan Orde Filter

Untuk menentukan orde filter yang optimal dilakukan pengujian terhadap data yang didapat dari pembacaan akselerometer. Tabel 4.10 menunjukkan koefisien rekursi untuk tiap orde dari orde 1 hingga orde 10 pada filter Bessel dengan frekuensi *sampling* 125 Hz dan frekuensi *cut off* 5 Hz.

Setelah mendapatkan persamaan rekursi dan gain dari tiap – tiap orde, langkah selanjutnya adalah melakukan pengambilan data.

Data yang diambil adalah data pergerakan pada sumbu x sejauh -30 cm. Data yang diambil sebanyak 10 kali kemudian dari sekumpulan data percobaan dilakukan integrasi kedua untuk memperoleh data

perpindahan. Dari data perpindahan dapat diambil nilai error dalam rMSE.

Orde filter yang dipilih adalah orde filter yang memberikan nilai rMSE paling kecil. Hasil pengujian filter tiap orde ditunjukkan oleh tabel 4.11 dibawah. Sedangkan grafik data percepatan sebelum dan sesudah menggunakan filter dari orde 1 hingga 10 ditunjukkan oleh gambar 4.29.

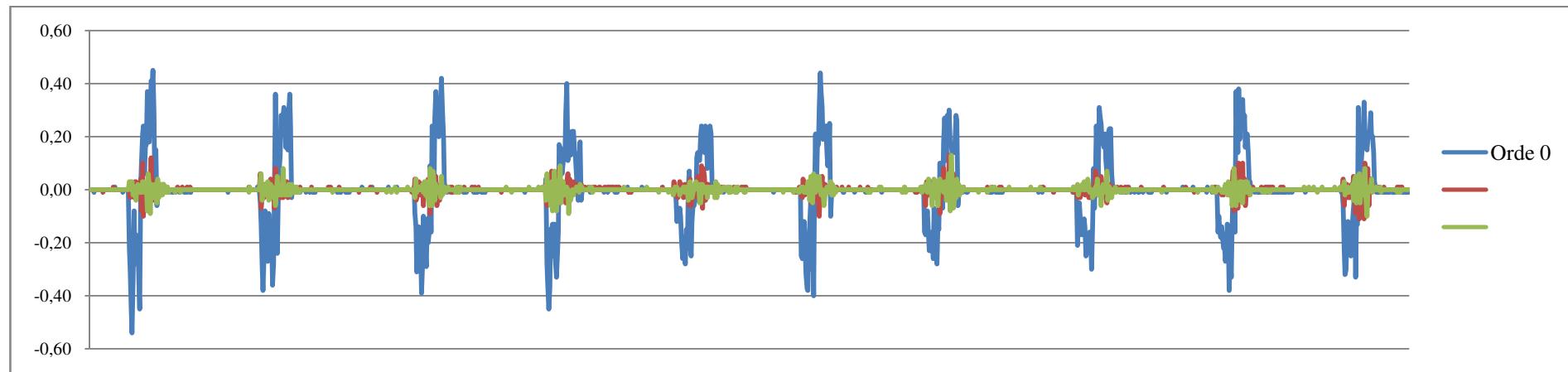
Tabel 4.10 Koefisien rekursi filter besel orde 1 hingga 10 dengan frekuensi *cut off* 5 Hz dan frekuensi *sampling* 125 Hz

Orde	gain	Koefisien Rekursi			
1	8.916	a0 = 1 a1 = 1	b1 = 0.776		
2	5.051e+01	a0 = 1 a1 = 2 a2 = 1	b1 = -0.573 b2 = 1.494		
3	2.711e+02	a0 = 1 a1 = 3 a2 = 3 a3 = 1	b1 = 0.423 b2 = -1.655 b3 = 2.203		
4	1.331e+03	a0 = 1 a1 = 4 a2 = 6 a3 = 4 a4 = 1	b1 = -0.304 b2 = 1.596 b3 = -3.191 b4 = 2.887		
5	5.908e+03	a0 = 1 a1 = 5 a2 = 10 a3 = 10 a4 = 5 a5 = 1	b1 = 0.212 b2 = -1.399 b3 = 3.753 b4 = -5.110 b5 = 3.539		
6	2.396e+04	a0 = 1 a1 = 6 a2 = 15 a3 = 20 a4 = 15 a5 = 6 a6 = 1	b1 = 0.142 b2 = 1.139 b3 = -3.851 b4 = 7.038 b5 = -7.346 b6 = 4.159		
7	9.019e+04	a0 = 1 a1 = 7 a2 = 21 a3 = 35 a4 = 35 a5 = 21 a6 = 7 a7 = 1	b1 = 0.093 b2 = -0.874 b3 = 3.578 b4 = -8.239 b5 = 11.543 b6 = 9.851 b7 = 4.750		
8	3.194e+05	a0 = 1			

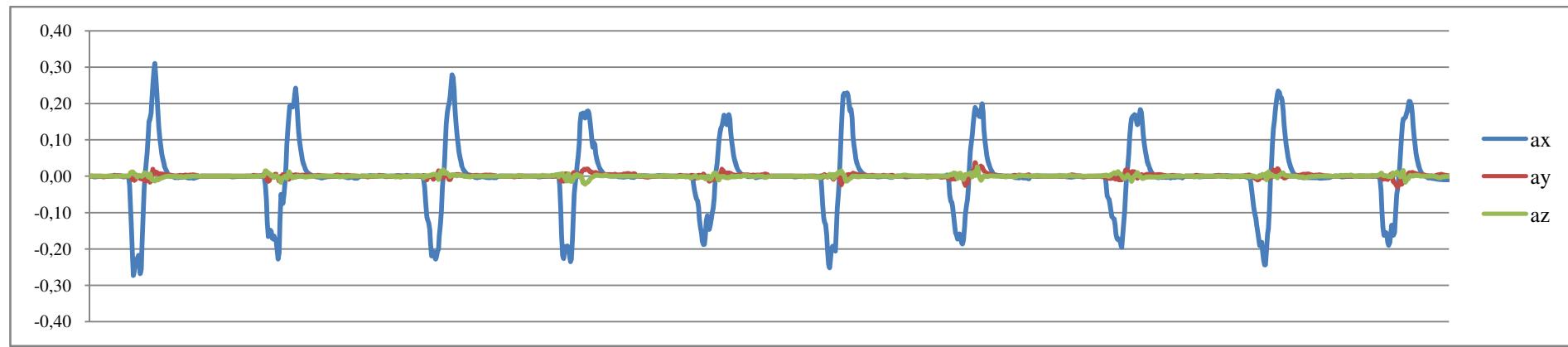
		a1 = 8 a2 = 28 a3 = 56 a4 = 70 a5 = 56 a6 = 28 a7 = 8 a8 = 1	b1 = 0.059 b2 = 0.6389 b3 = -3.0779 b4 = 8.574 b5 = -15.124 b6 = 17.317 b7 = -12.583 b8 = 5.313
9	1.074e+06	a0 = 1 a1 = 9 a2 = 36 a3 = 84 a4 = 126 a5 = 126 a6 = 84 a7 = 36 a8 = 9 a9 = 1	b1 = 0.036 b2 = -0.447 b3 = 2.484 b4 = -8.141 b5 = 17.366 b6 = -25.022 b7 = 24.379 b8 = -15.508 b9 = 5.852
10	3.457e+06	a0 = 1 a1 = 10 a2 = 45 a3 = 120 a4 = 210 a5 = 252 a6 = 210 a7 = 120 a8 = 45 a9 = 10 a10 = 1	b1 = -0.028 b2 = 0.301 b3 = -1.900 b4 = 7.176 b5 = -17.997 b6 = 31.333 b7 = -38.393 b8 = 32.726 b9 = -18.595 b10 = 6.368

Tabel 4.11 Nilai rMSE pada pengujian filter dari orde 1 hingga 10 dan data yang tidak difilter.

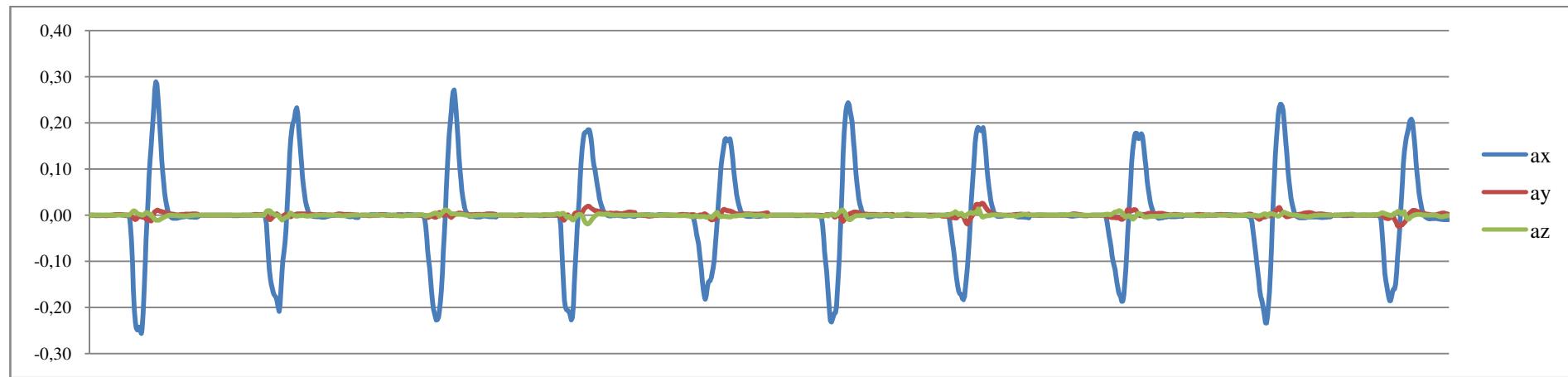
Filter	Percobaan ke -										rMSE
	1	2	3	4	5	6	7	8	9	10	
Tidak difilter	-0.255	-0.386	-0.276	-0.239	-0.309	-0.368	-0.340	-0.337	-0.348	-0.355	0.0474
Orde 1	-0.256	-0.381	-0.279	-0.243	-0.309	-0.364	-0.339	-0.335	-0.347	-0.352	0.0450
Orde 2	-0.257	-0.379	-0.280	-0.244	-0.309	-0.363	-0.339	-0.335	-0.346	-0.351	0.0441
Orde 3	-0.257	-0.377	-0.281	-0.246	-0.310	-0.362	-0.339	-0.334	-0.345	-0.350	0.0432
Orde 4	-0.258	-0.375	-0.283	-0.248	-0.310	-0.361	-0.339	-0.333	-0.345	-0.349	0.0423
Orde 5	-0.259	-0.374	-0.284	-0.249	-0.310	-0.360	-0.340	-0.333	-0.344	-0.349	0.0416
Orde 6	-0.260	-0.372	-0.285	-0.251	-0.310	-0.359	-0.340	-0.332	-0.344	-0.348	0.0409
Orde 7	-0.261	-0.371	-0.286	-0.252	-0.310	-0.358	-0.340	-0.332	-0.344	-0.347	0.0404
Orde 8	-0.262	-0.370	-0.286	-0.253	-0.310	-0.357	-0.340	-0.332	-0.344	-0.347	0.0398
Orde 9	-0.262	-0.369	-0.287	-0.254	-0.310	-0.356	-0.340	-0.332	-0.343	-0.346	0.0393
Orde 10	-0.263	-0.369	-0.288	-0.255	-0.310	-0.356	-0.340	-0.331	-0.343	-0.346	0.0389



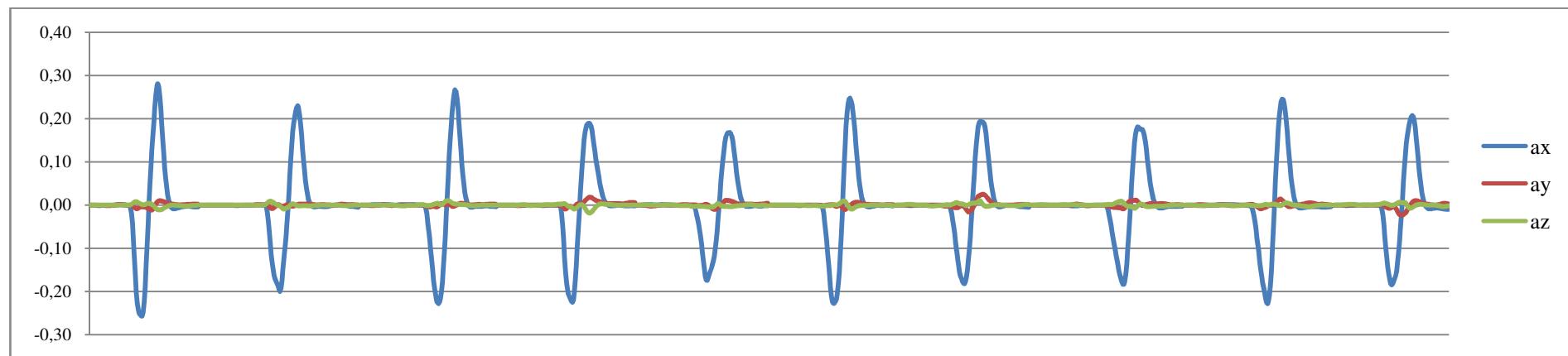
a. Tidak di filter



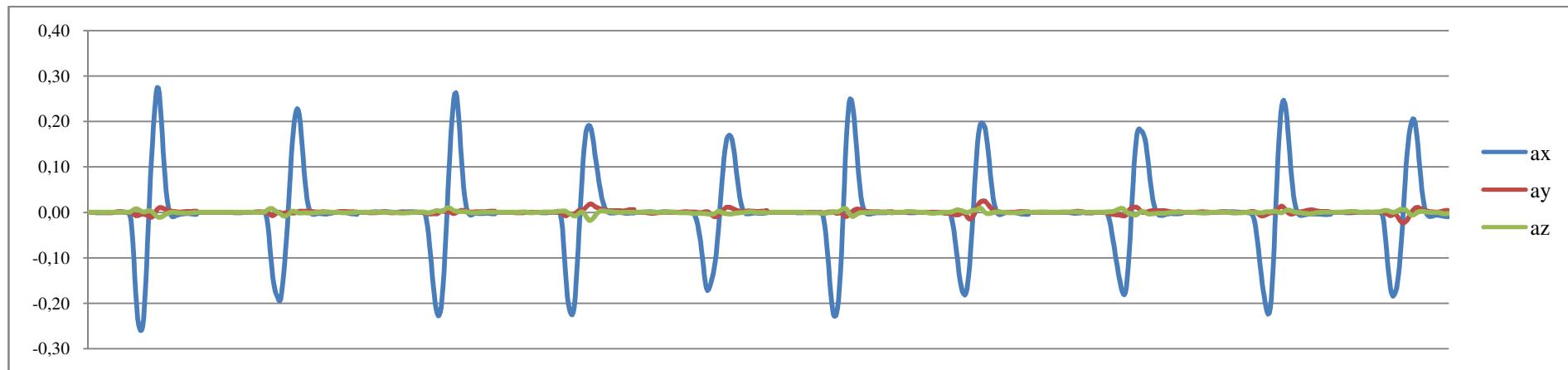
b. Orde 1



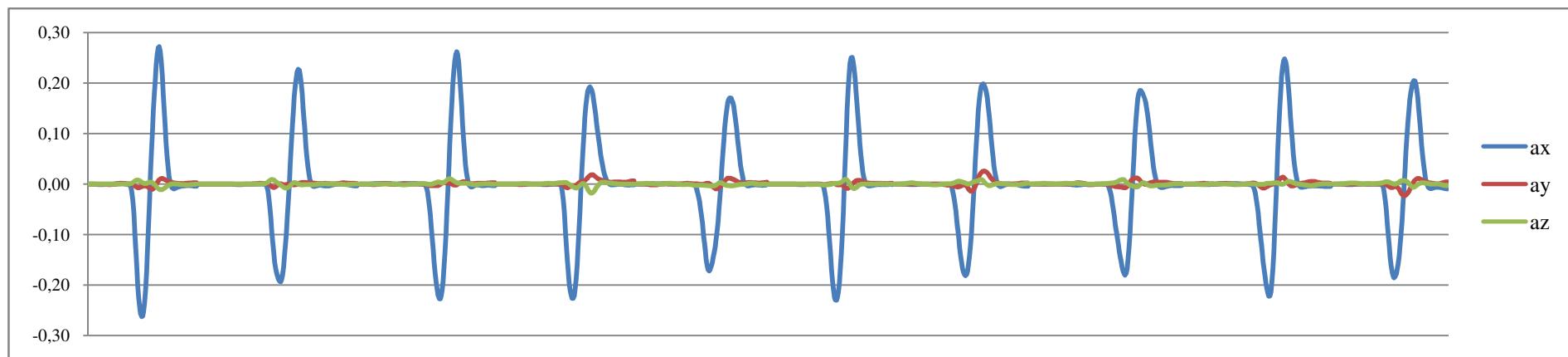
c. Orde 2



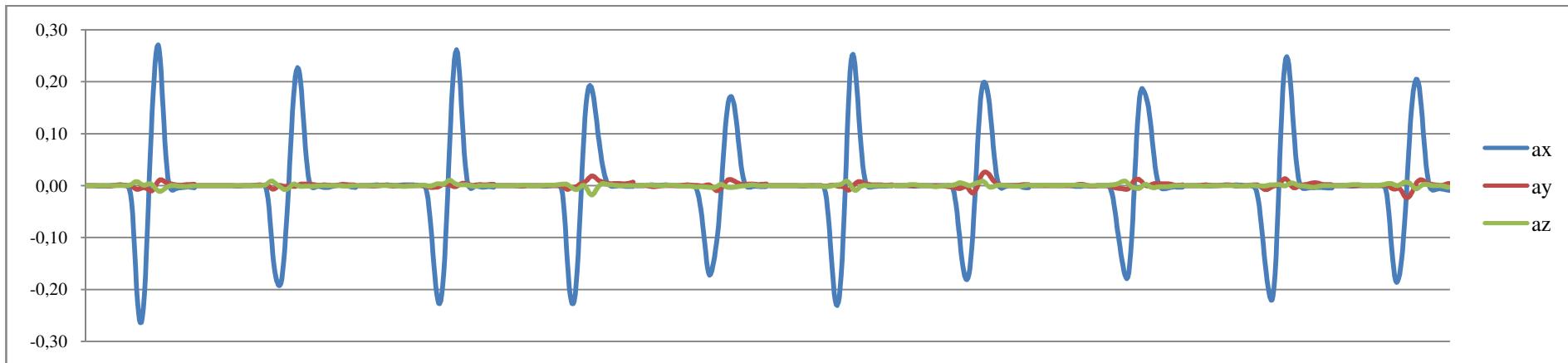
d. Orde 3



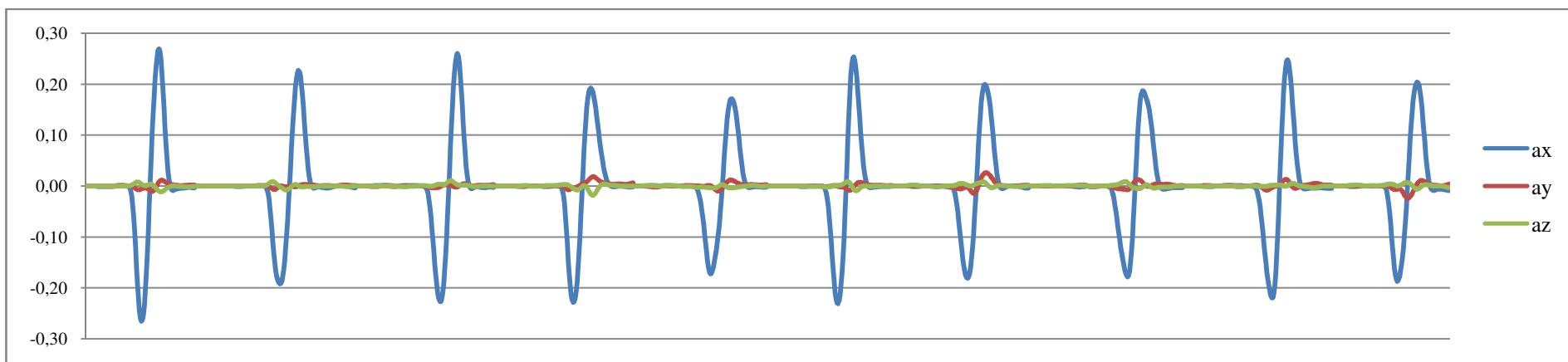
e. Orde 4



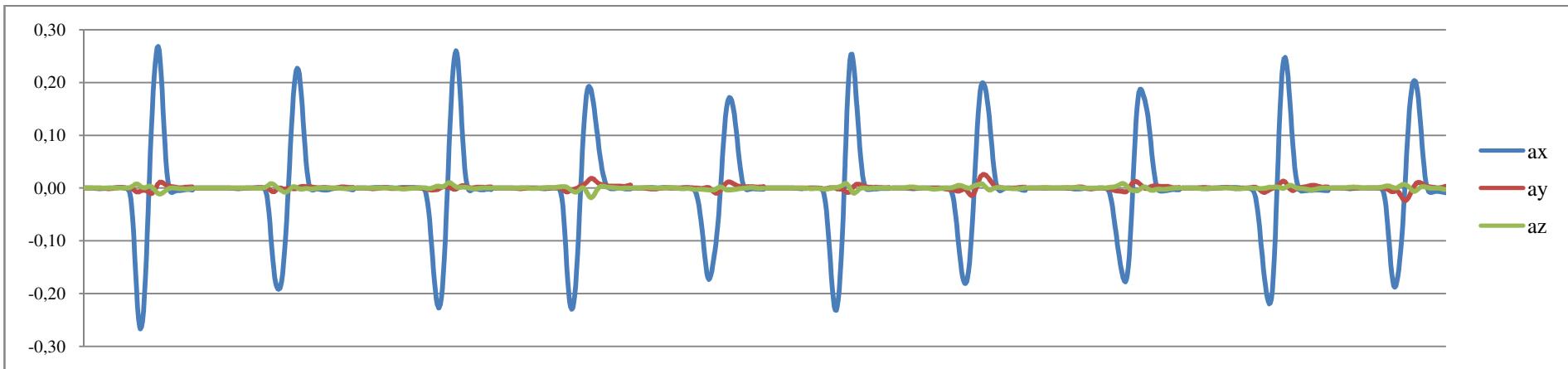
f. Orde 5



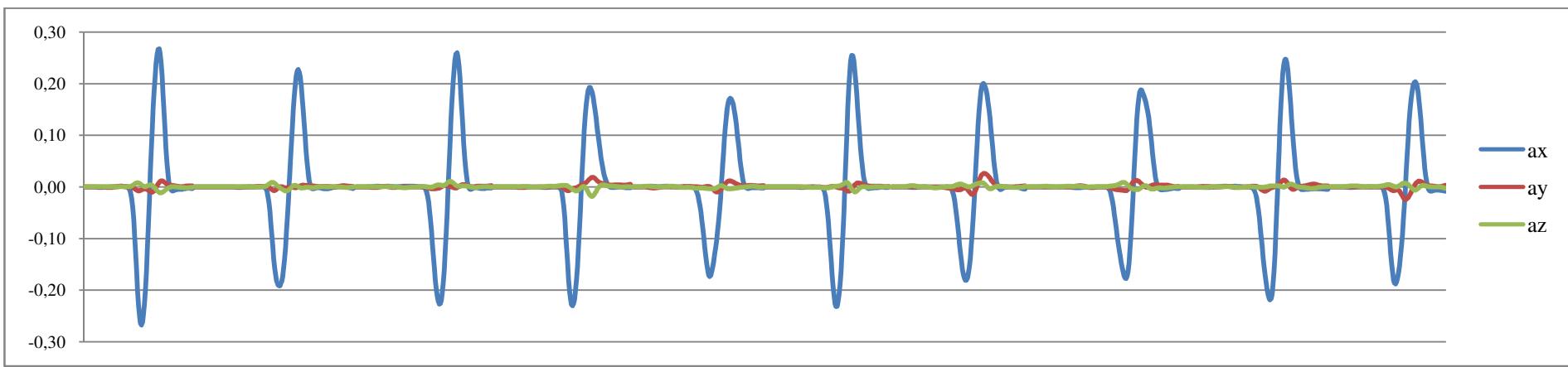
g. Orde 6



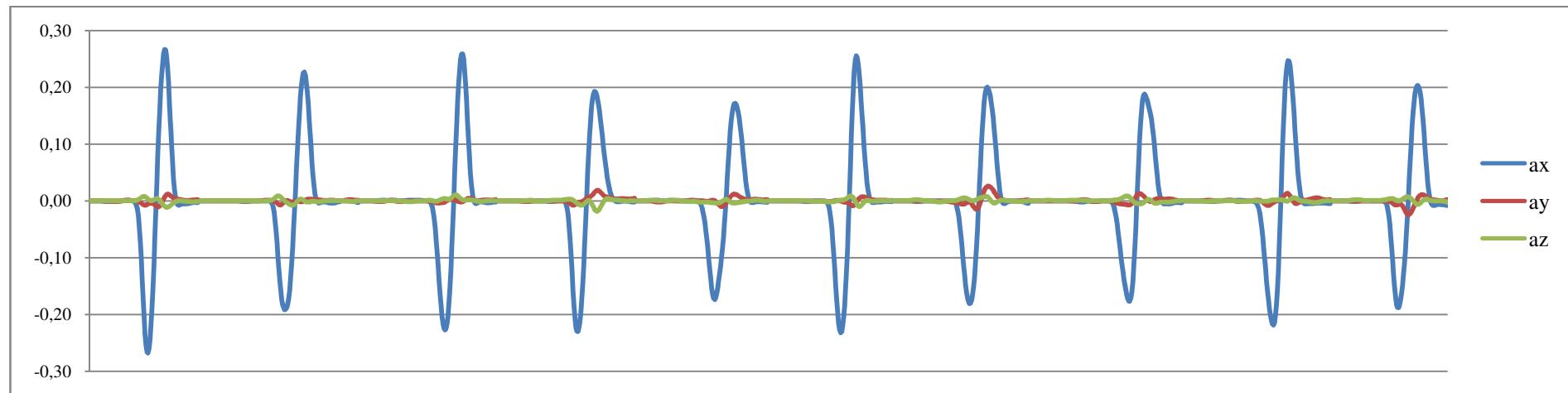
h. Orde 7



i. Orde 8



j. Orde 9

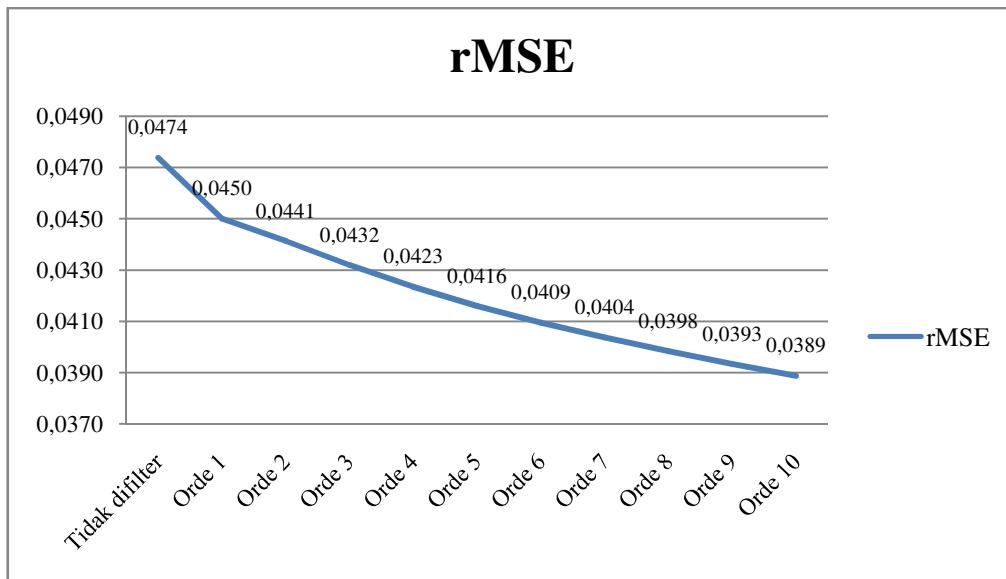


k. Orde 10

Gambar 4.29 Grafik nilai percepatan sebelum dan sesudah menggunakan filter orde 1 hingga 10

Grafik nilai rMSE ditunjukan oleh gambar 4.30 dibawah.

Tampak bahwa filter orde 10 memberikan nilai rMSE yang paling kecil.



Gambar 4.30 Grafik nilai rMSE hasil integrasi kedua (data perpindahan) pada tiap – tiap orde filter dan data yang tidak menggunakan filter.

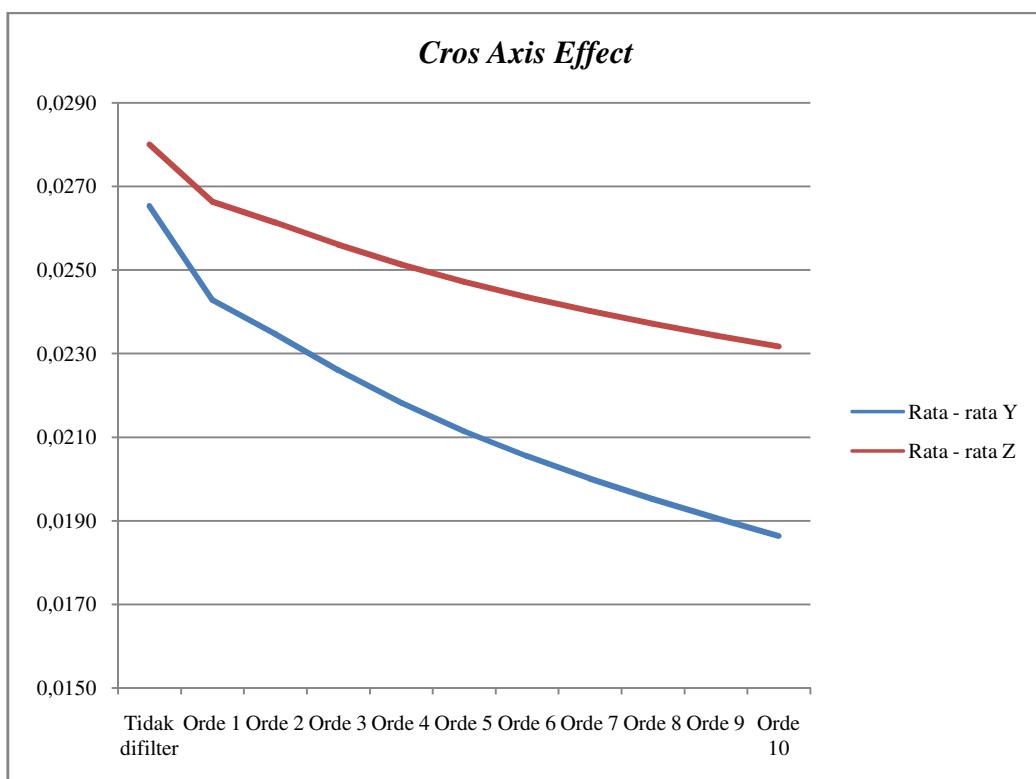
Selain itu dapat diamati juga *cross axis effect* dari setiap percobaan. Tabel 4.12 menunjukan nilai rata – rata *cross axis effect* pada pengujian tidak menggunakan filter dan menggunakan filter orde 1 hingga 10 sedangkan grafiknya ditunjukan oleh gambar 4.31.

Dari hasil yang didapat tampak bahwa filter orde 10 memberikan nilai rata – rata pergerakan pada sumbu Y dan sumbu Z paling kecil. Sehingga dapat disimpulkan bahwa *cross axis effect* pada penggunaan filter orde 10 paling kecil.

Berdasarkan nilai rMSE dan *cross axis effect* maka dipilih filter dengan orde 10.

Tabel 4.12 *Cross Axis Effect* pada pengujian tidak menggunakan filter dan menggunakan filter orde 1 hingga 10.

Filter	Rata - rata Y	Rata - rata Z
Tidak difilter	0.0265	0.0280
Orde 1	0.0243	0.0266
Orde 2	0.0235	0.0261
Orde 3	0.0226	0.0256
Orde 4	0.0218	0.0251
Orde 5	0.0211	0.0247
Orde 6	0.0205	0.0244
Orde 7	0.0200	0.0240
Orde 8	0.0195	0.0237
Orde 9	0.0191	0.0234
Orde 10	0.0186	0.0232



Gambar 4.31 Grafik cross axis effect pada pengujian tidak menggunakan filter dan menggunakan filter orde 1 hingga 10

b. Penerapan Pada Algoritma Program

Pada tabel 4.10 ditunjukan persamaan rekursi dari fungsi alih filter yang dihasilkan dengan $x[n]$ adalah sebagai input dan $y[n]$ sebagai output. Langkah selanjutnya adalah menerapkanya kedalam program arduino.

Berikut fungsi filter IIR pada arduino yang diigunakan.

```
// LPF Bessel, fc = 5 Hz Fs = 125 Hz Orde=10
#define GAIN 3.457215320e+06

float IIR(float value, float xv[], float yv[]) {
    xv[0] = xv[1];
    xv[1] = xv[2];
    xv[2] = xv[3];
    xv[3] = xv[4];
    xv[4] = xv[5];
    xv[5] = xv[6];
    xv[6] = xv[7];
    xv[7] = xv[8];
    xv[8] = xv[9];
    xv[9] = xv[10];
    xv[10] = value / GAIN;

    yv[1] = yv[2];
    yv[2] = yv[3];
    yv[3] = yv[4];
    yv[4] = yv[5];
    yv[5] = yv[6];
    yv[6] = yv[7];
    yv[7] = yv[8];
    yv[8] = yv[9];
    yv[9] = yv[10];
    yv[10] = (xv[0]+xv[10]) + 10*(xv[1]+xv[9]) + 45*(xv[2]+xv[8]) +
              120*(xv[3]+xv[7]) + 210*(xv[4]+xv[6]) + 252*xv[5] +
              (-0.0217192171*yv[0]) + (0.3012846437*yv[1]) +
              (-1.8998620952*yv[2]) + (7.1765621146*yv[3]) +
              (-17.9969646061*yv[4]) + (31.3334899657*yv[5]) +
              (-38.3927239217*yv[6]) + (32.7257173947*yv[7]) +
              (-18.5945171308*yv[8]) + (6.3684366602*yv[9]);

    return yv[10];
}
```

Untuk memanggil fungsi IIR diatas dilakukan dengan cara sebagai berikut.

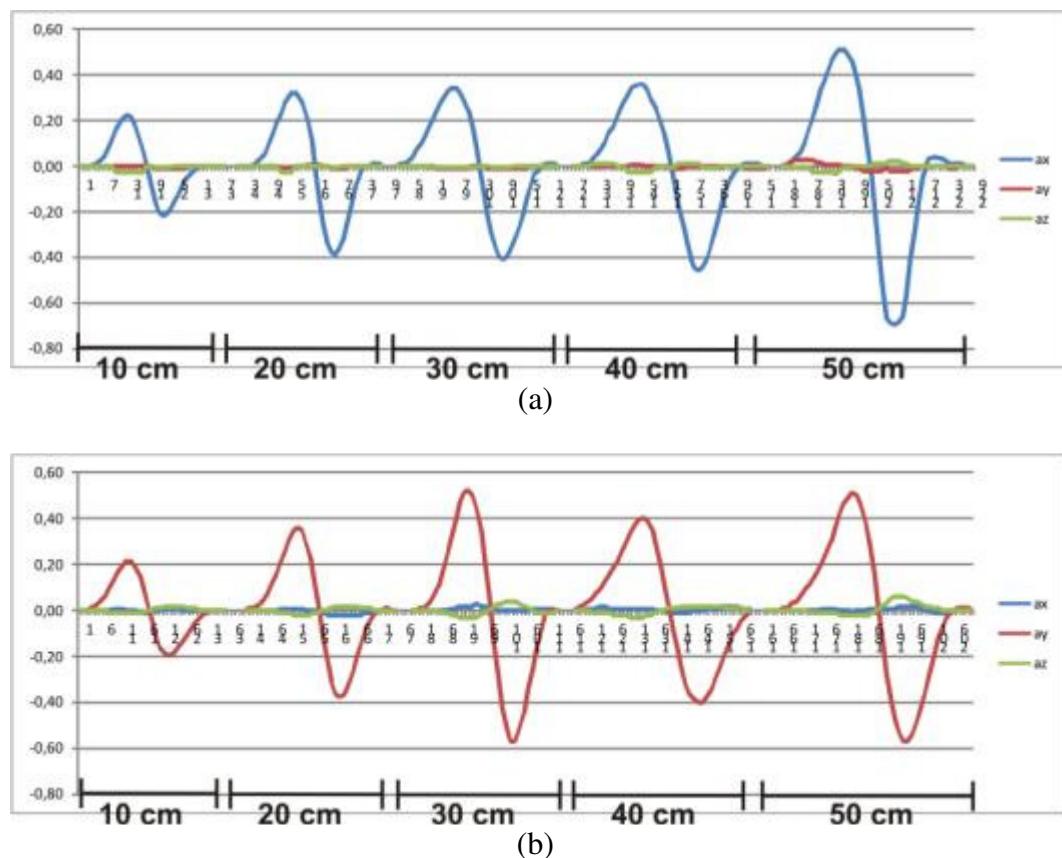
```
//===== Filtering data akselerometer=====/
float accelXIn[11], accelXIn[11], accelXIn[11];
float accelXOut[11], accelXOut[11], accelXOut[11];

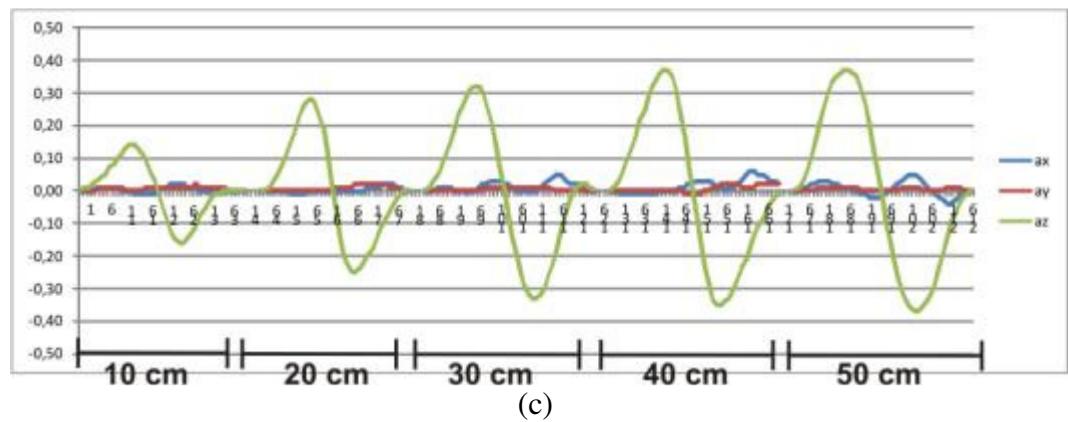
float accelX = IIR(Ax, accelXIn, accelXOut);
float accelY = IIR(Ay, accelYIn, accelYOut);
float accelZ = IIR(Az, accelZIn, accelZOut);
```

1. Hasil Pengukuran Setelah Menggunakan Filter

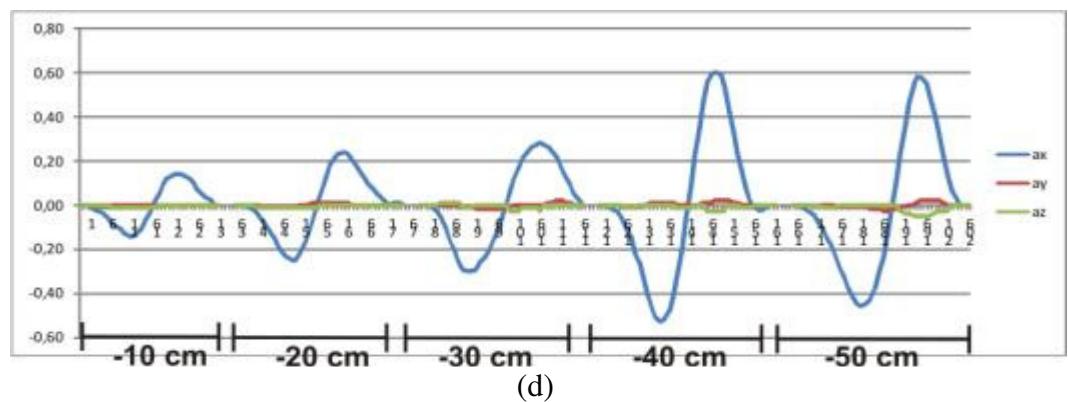
Pengukuran kembali dilakukan dengan metode mengukur percepatan pada masing-masing sumbu dengan arah pergerakan positif dan negatif untuk mengetahui efek penggunaan filter pada modul.

Hasil yang didapat pada pengukuran sumbu X, Y, dan Z pada arah positif maupun negatif ditunjukkan oleh gambar 4.32 berikut.

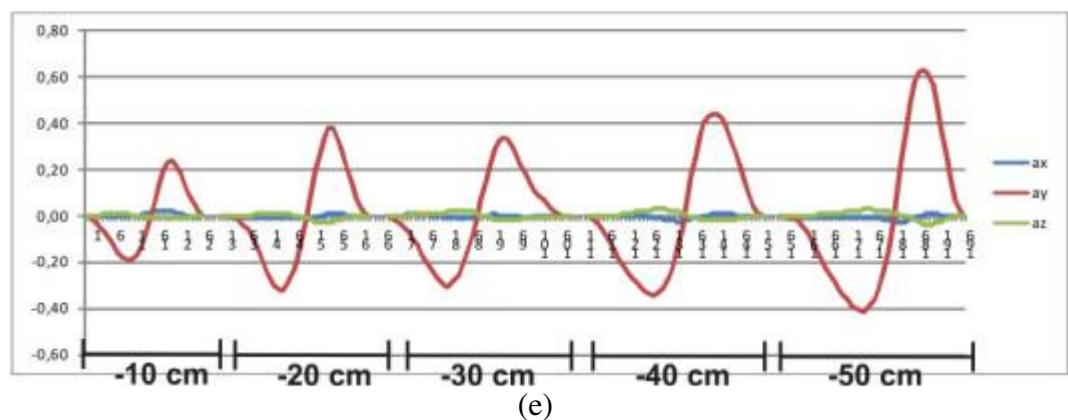




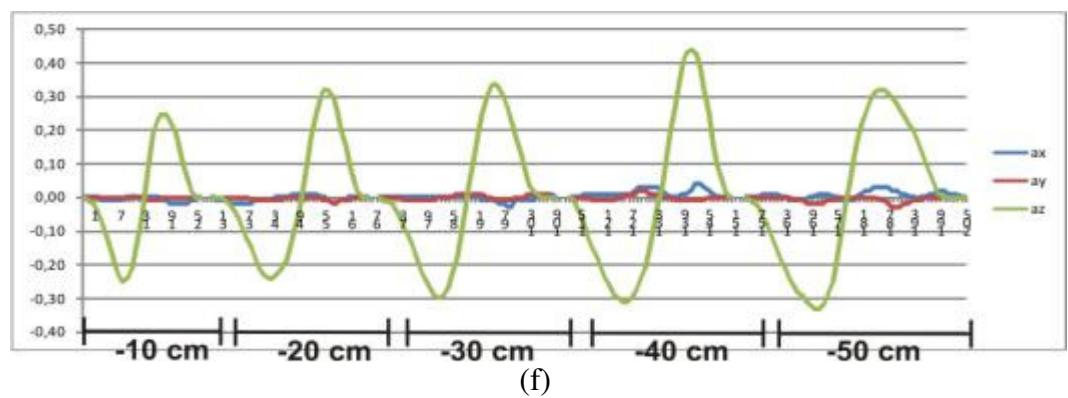
(c)



(d)



(e)



(f)

Gambar 4.32 Hasil pengujian akselerometer setelah difilter (a) Sumbu x arah positif, (b) Sumbu x arah positif, (c) Sumbu z arah positif, (d) Sumbu x arah negatif, (e) Sumbu y arah negatif, (f) Sumbu z arah negatif

Berdasarkan grafik yang ditunjukan oleh gambar 4.32 diatas tampak bahwa hasil yang didapatkan setelah menerapkan filter pada akusisi data akselerometer menunjukan kurva pergerakan lebih halus dan lebih baik. Tampak bahwa noise pada tiap – tiap sumbu dapat dikurangi. Hal ini menunjukan bahwa filter bekerja sesuai dengan yang diharapkan.

Langkah selanjutnya adalah melakukan double integral untuk mendapatkan data jarak perpindahan. Untuk hasil masing – masing pergerakan sumbu, hasil double integral ditunjukan oleh tabel 4.13 dibawah.

Tabel 4.13 Data hasil pengujian pergerakan akselerometer pada sumbu x, y, dan z setelah difilter

Jarak (cm)	X			Y			Z		
	Terukur	Error	Akurasi	Terukur	Error	Akurasi	Terukur	Error	Akurasi
10	11.02	1.02	89.8%	11.24	1.24	87.6%	11.75	1.75	82.5%
20	20.03	0.03	99.9%	20.94	0.94	95.3%	20.79	0.79	96.1%
30	33.75	3.75	87.5%	31.99	1.99	93.4%	32.29	2.29	92.4%
40	45.08	5.08	87.3%	43.09	3.09	92.3%	42.71	2.71	93.2%
50	54.1	4.1	91.8%	51.95	1.95	96.1%	53.18	3.18	93.6%
-10	-9.61	-0.39	96.1%	-10.43	0.43	95.7%	-11.45	1.45	85.5%
-20	-19.7	-0.3	98.5%	-21.81	1.81	91.0%	-21.8	1.8	91.0%
-30	-31.79	1.79	94.0%	-30.41	0.41	98.6%	-29.56	-0.44	98.5%
-40	-43.04	3.04	92.4%	-42.74	2.74	93.2%	-39.92	0.08	99.8%
-50	-50.1	0.1	99.8%	-54.22	4.22	91.6%	-53.1	3.1	93.8%
rMSE	2.65			2.21			2.04		

Sedangkan pengaruh pergerakan terhadap sumbu yang tidak digerakan ditunjukkan oleh tabel 4.14 berikut.

Tabel 4.14 Hasil *double integrasi* data percepatan pada sumbu yang tidak bergerak setelah difilter

Jarak (cm)	Pergeseran Sumbu X		Pergeseran Sumbu Y		Pergeseran Sumbu Z	
	Y	Z	X	Z	X	Y
10	-1.19	-5.47	0.45	-0.16	1.90	3.63
20	-0.55	-2.17	-0.87	-1.00	-0.33	1.83
30	-4.81	-4.69	2.67	-1.63	6.17	2.16
40	-0.21	-3.45	4.04	-3.50	3.94	1.39
50	3.22	-6.91	2.52	0.62	5.34	2.74
-10	-0.53	-1.45	0.85	0.15	-1.34	-0.31
-20	-0.68	-2.64	-2.61	-0.53	-3.52	-1.69
-30	-2.64	-1.03	-0.47	1.84	-1.55	-0.97
-40	1.07	-2.79	-2.35	2.69	5.64	-0.33
-50	-3.19	-4.78	-5.65	3.01	3.03	-5.66

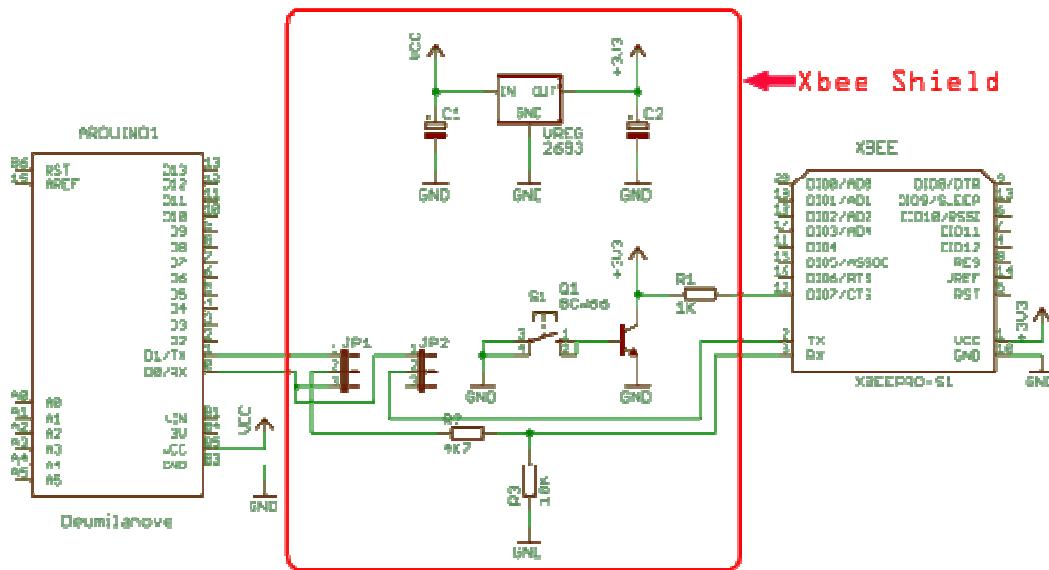
Jika dibandingkan dengan data perhitungan yang belum menggunakan filter, data hasil perhitungan yang menggunakan filter jauh lebih baik walaupun pengaruh gravitasi dan *cross axis efect* belum bisa dihilangkan.

IV.3 Unit Komunikasi (modul RF)

Unit komunikasi merupakan bagian yang bertugas untuk mengirim dan menerima data dari tiap *node*. Dalam pembuatan jaringan sensor nirkabel ini terdapat 3 buah *node* yang terdiri dari 2 buah *router* dan 1 buah *gateway*. Antara *router* dan *gateway* memiliki perbedaan skematik dalam hal jalur komunikasi dengan mikrokontroler serta perbedaan algoritma yang digunakan. Sehingga dalam unit komunikasi dibagi menjadi dua buah yaitu *router* dan *gateway*.

IV.3.1 Skematik rangkaian modul RF pada router

Skematik rangkaian perantaraan modul RF (Xbee Pro Series 1) dengan mikrokontroler (Arduino Deumilanove) ditunjukan oleh gambar 4.33.



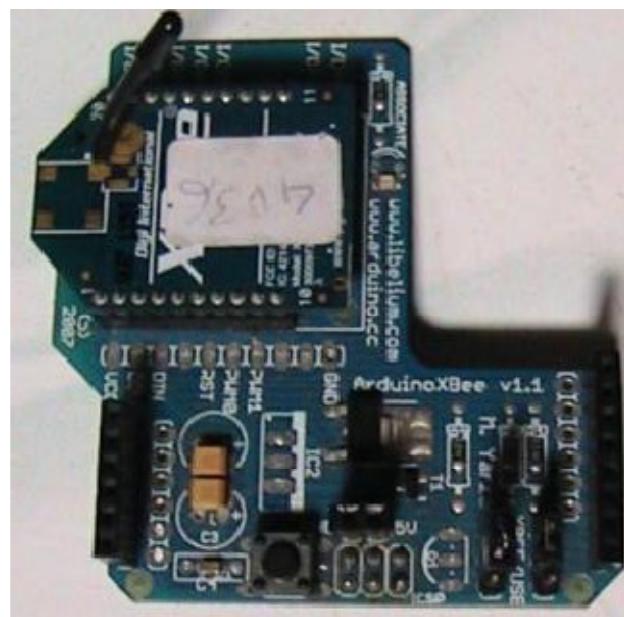
Gambar 4.33 Skematik perantaraan Xbee Pro Series 1 dengan Arduino Deumilanove pada router

Pada router digunakan *Xbee Shield* (kotak merah) sebagai perantara antara arduino dengan modul Xbee. Penggunaan *Xbee Shield* memberikan keuntungan sebagai berikut.

1. Instalasi xbee dengan arduino dipermudah karena tinggal menancapkan modul pada soket yang tersedia.
2. Tersedia regulator tegangan 3,3 Volt yang sesuai dengan tegangan suplai yang dibutuhkan modul Xbee sehingga tidak perlu membuat penyesuaian tegangan lagi.

3. Dapat digunakan sebagai perantara untuk pemrograman Xbee dengan cara melepas mikrokontroler yang ada pada modul arduino.
4. Terdapat penyesuaian level tegangan antara pin TX arduino dengan pin RX Xbee.

Bentuk *Xbee Shield* beserta Xbee Pro Series 1 yang telah terpasang pada *Xbee Shield* ditunjukkan oleh gambar 4.34.



Gambar 4.34. Xbee Shield yang telah terpasang modul Xbee Pro Series 1.

Pada *router*, jalur komunikasi antara modul Xbee dengan arduino sama dengan jalur komunikasi yang digunakan untuk komunikasi serial arduino dengan komputer melalui perantara konverter *USB to Serial* menggunakan FT232. Sehingga apabila *Xbee Shield* terpasang pada arduino dan akan melakukan proses *download* program dari komputer ke arduino harus mengubah posisi sambungan pada

jumper JP2 (gambar 4.8) yakni pin nomor 2 dihubungkan dengan pin no 3. Sehingga hubungan pin RX dari arduino dengan pin TX Xbee terputus dan program dapat ter-*download* dengan sempurna. Jika tidak dilakukan pemindahan posisi jumper JP2 maka akan menyebabkan *error* pada proses penulisan program dari komputer ke arduino.

IV.3.2 Algoritma Pengiriman Data Pada Router

Pembuatan algoritma program pengiriman data pada *router* menggunakan bantuan *library* untuk xbee (file “Xbee.h”) yang ditulis oleh Andrew Rapp dari <http://fsf.org/> (2009). Sebelum membuat program terlebih dahulu dilakukan seting parameter pada modul Xbee sebagai berikut.

Parameter	Nilai
BD (<i>Baud rate</i>)	9600
DL (<i>Destination Address Low</i>)	7E63
AP (<i>API mode</i>)	2

Setelah parameter di setting sesuai tabel diatas, langkah selanjutnya adalah penyusunan algoritma program. Pembuatan algoritma *router* mengacu pada *library* yang digunakan serta penyesuaian dengan karakteristik modul Xbee. Langkah – langkah pembuatan algoritma *router* adalah sebagai berikut.

1. Memasukan *library* Xbee.h kedalam program.
2. Membuat variabel bertipe Xbee.

3. Membuat variabel *buffer* untuk menampung data yang akan dikirim. Variabel ini bertipe integer positif 8 bit (1 byte) dan berbentuk *array* dengan besarnya *array* tergantung dari banyaknya data yang akan dikirim.
4. Membuat variabel yang bertipe *XbeeAddress64* dan nilainya diisi dengan *XbeeAddress64(DH,DL)* dimana DH merupakan 32 bit MSB dari alamat Xbee yang dituju dan DL merupakan 32 bit LSB dari alamat Xbee yang dituju.
5. Membuat variabel bertipe *Tx16Request* untuk menampung data keseluruhan yang akan dikirim yang mana variabel ini akan diisi dengan hasil dari fungsi *Tx16Request(addr16,payload, sizeof(payload))*. *Addr16* merupakan 16 bit LSB alamat pengirim.
6. Membuat variabel bertipe *TxStatusResponse*. Variabel ini akan menampung hasil dari fungsi *TxStatusResponse()*.
7. Membuat algoritma pengiriman data yakni mengisi *payload* yang telah dibuat dengan data yang akan dikirim.

Pada program pengiriman data pada *router* dibuat variabel dengan nama *xbee* yang bertipe *Xbee* kemudian variabel *buffer* untuk menampung data yang akan dikirim bernama *payload* dengan tipe *uint8_t* (*unsigned integer 8 bit*) atau 1 byte dan berbentuk *array* dengan panjang *array* 13. Panjang *array* ditentukan dengan menghitung

banyaknya *byte* data yang akan dikirim. Data – data yang akan dikirim diantaranya adalah sebagai berikut.

1. Data dari unit RTC berupa data jam, menit, detik, tanggal, bulan dan tahun yang masing – masing 1 *byte* sehingga panjang data dari RTC berjumlah 6 *byte*.
2. Data jarak perpindahan modul hasil integrasi data percepatan yang dibaca oleh akselerometer. Data jarak perpindahan terdiri dari 3 sumbu yang tiap – tiap sumbu terdiri dari 2 *byte* sehingga jumlah data yang dikirim sebanyak 6 *byte*. Alasan mengapa data perpindahan membutuhkan alokasi 2 *byte* dikarenakan data perpindahan ada yang bernilai negatif sedangkan mengirimkan bilangan negatif pada komunikasi serial sering menimbulkan masalah pada perhitungan sehingga *payload* bertipe `uint8_t` yang memiliki jangkauan nilai dari 0 hingga 255. Agar dapat mengirimkan data negatif maka ditambahkan 1 *byte* lagi sebagai penanda data positif atau negatif. Jika data positif , *byte* ini akan bernilai 0x00 sedangkan jika datanya negatif, *byte* ini bernilai 0xff. Data perpindahan sendiri bertipe `float` dengan ketelitian 2 angka dibelakang koma sehingga agar data perpindahan dapat dikirim maka dilakukan konversi ke `uint8_t` dengan dikalikan 100 terlebih dahulu.
3. Jumlah keseluruhan data yang akan dikirim adalah 12 *byte* yakni 6 *byte* data dari RTC dan 6 *byte* data jarak perpindahan.

Dengan mengirimkan data sebanyak 12 *byte* maka panjang data (*data length*) yang diterima oleh *gateway* seharusnya adalah 17 *byte* karena mendapat tambahan 5 *byte* yang terdiri dari 1 *byte* data *API identifier*, 2 *byte* data *ID router*, 1 *byte* data RSSI (*Ricieve Signal Strength*) dan 1 *byte* data *options byte*. Desimal 17 dalam format heksadesimal adalah 0x11.

4. Dalam format API (*Aplication Programming Interface*) mode 2 terdapat *escape byte* yakni *byte – byte* yang berfungsi sebagai karakter khusus dalam format ASCII diantaranya 0x7E (karakter ~ / *start delimiter* pada jaringan Zigbee), 0x11(karakter Xon) dan 0x13 (karakter Xoff / enter). Sehingga jika *data length* adalah 0x11 maka akan terdeteksi sebagai *escape byte* . Salah satu cara untuk menghindari *escape byte* adalah dengan menambahkan 1 *byte* data pada *payload* sehingga jumlah *payload* menjadi 13 dan *data length* menjadi 0x12.

Setelah menentukan format data yang dikirim, langkah selanjutnya adalah menentukan periode pengiriman data.

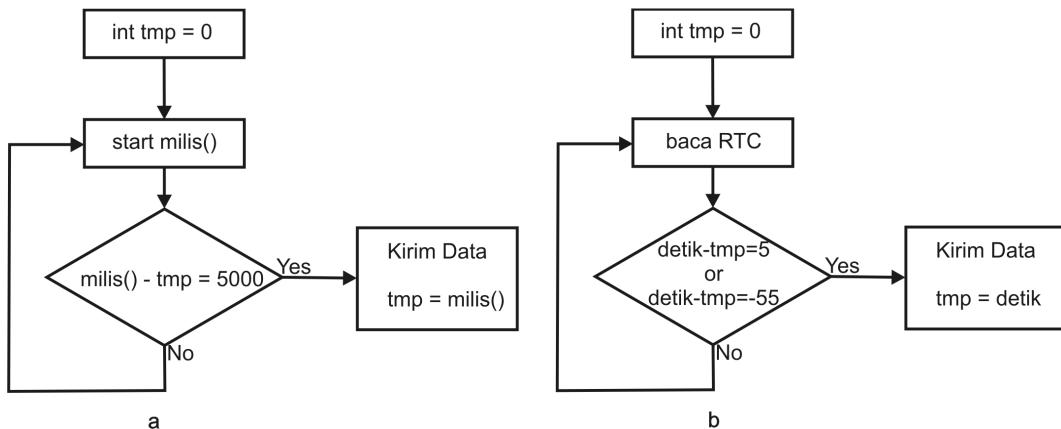
Untuk menentukan periode pengiriman data dapat dilakukan dengan dua alternatif yaitu memanfaatkan *timer/counter* internal yang dimiliki oleh mikrokontroler atau dengan memanfaatkan *timer* eksternal yaitu dari modul RTC.

Jika menggunakan *timer/counter* internal mikrokontroler pada arduino dapat memanfaatkan fungsi `milis()` atau `micros().fungsi`

`milis()` merupakan fungsi penghitung naik (*count up*) dengan nilai keluaran bertipe `unsigned long` sehingga jangkauan *counter* ini adalah dari 0 hingga 4.294.967.295 atau 4 byte ($2^{32} - 1$). Nilai *counter* akan bertambah satu setiap 1 milidetik sehingga nilai maksimal 4.294.967.295 akan tercapai setelah 49 hari lebih 17 jam 2 menit 47 detik setelah arduino dijalankan. Dan pada saat ini *counter* akan mengalami *over flow* dan kembali ke – 0. Sedangkan fungsi `micros()` merupakan fungsi yang serupa dengan `milis()` akan tetapi nilai *counter* bertambah setiap 1 mikrodetik. Fungsi ini juga bertipe `unsigned long` sehingga akan mengalami *over flow* setelah 71 menit 35 detik arduino dijalankan.

Alternatif pewaktuan kedua adalah dengan memanfaatkan data yang diberikan oleh unit RTC. Data yang diberikan oleh RTC berupa data tahun, bulan, tanggal, hari, jam, menit dan detik. Untuk mengatur selang waktu pengiriman data dapat memanfaatkan data waktu ini.

Perbedaan algoritma perhitungan selang waktu pengiriman data menggunakan *timer/counter* internal dengan *timer* dari unit RTC ditunjukan pada flowchart pada gambar 4.35 dibawah. Kedua program sama - sama mengatur selang waktu pengiriman data tiap 5 detik.



Gambar 4.35 Flowchart algoritma pewaktuan pengiriman data. (a) menggunakan counter internal, (b) menggunakan RTC

Kelemahan dari pewaktuan menggunakan *counter* internal yakni pada akurasi pewaktuan. Karena arduino menggunakan crystal 16 MHz tingkat akurasi *counter* sebesar 4 mikrodetik sehingga tiap penambahan nilai maka *error* sebesar 4 mikrodetik ini ikut bertambah juga. *Counter* internal ini juga akan mengalami gangguan jika mikrokontroler mengalami *hang* pada proses perhitungan ataupun proses transfer data antar register.

Sedangkan pemakaian unit RTC sebagai penentu waktu pengiriman data memiliki tingkat akurasi yang lebih baik selama tidak ada gangguan eksternal pada rangkaian. Penggunaan RTC eksternal juga lebih stabil karena sumber *clock* terpisah dari sistem mikrokontroler sehingga tidak terpengaruh oleh proses yang sedang terjadi pada internal mikrokontroler.

Agar pewaktuan pengiriman data berjalan konstan, proses perhitungan selang waktu pengiriman data dibuat dalam subroutines interupsi sehingga apabila nilai selang waktu telah didapat maka

program interupsi berjalan dan menunda proses yang sedang terjadi pada program utama.

Program utama pada router terdiri dari pengambilan data dari akselerometer, perhitungan jarak perpindahan serta menyimpan nilai – nilai variabel yang akan dikirim kedalam *payload*. Jadi ketika rutin interupsi berjalan, nilai yang dikirim merupakan nilai terakhir yang disimpan didalam *payload*. Setelah menjalankan rutin interupsi, nilai – nilai pada variabel yang dikirim tadi kemudian direset kembali dan *payload* diisi dengan nilai yang baru.

Program utama pengiriman data pada *router* dalam format bahasa C arduino adalah sebagai berikut.

```
void loop() {
    getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth,
    &month, &year);
    //=====routine interupsi=====
    noInterrupts();
    if((second-tmp == 5) || (second-tmp == -55)){
        xbee.send(tx);
        Vx, Vy, Vz, Dx, Dy, Dz = 0;
        flashLed(statusLed, 1, 300);
        tmp = second;
    }
    interrupts();
    //=====Data Waktu=====
    payload[0] = hour & 0xff;
    payload[1] = minute & 0xff;
    payload[2] = second & 0xff;
    payload[3] = dayOfMonth & 0xff;
    payload[4] = month & 0xff;
    payload[5] = year & 0xff;
    //=====Mengukur Percepatan=====
    Ax = ((float(getH48C(0,0) - 2048))*0.0022)-Xawal;
    Ay = ((float(getH48C(0,1) - 2048))*0.0022)-Yawal;
    Az = ((float(getH48C(1,0) - 2048))*0.0022)-Zawal;
    //=====remove vibrasi dengan filter IIR =====
    accelX = IIR(Ax, accelXInput, accelXOutput);
    accelY = IIR(Ay, accelYInput, accelYOutput);
    accelZ = IIR(Az, accelZInput, accelZOutput);
```

```

// ===== hitung kecepatan =====/
Vx += (accelX * 0.01);
Vy += (accelY * 0.01);
Vz += (accelZ * 0.01);
// ===== hitung perpindahan ===== /
Dx += (Vx * 0.01);
Dy += (Vy * 0.01);
Dz += (Vz * 0.01);
//=====konversi float ke uint8_t=====
if (Dx < 0 ){
    byteDx = byte(Dx*(-100));
    payload[6] = 0xff;
    payload[7] = byteDx & 0xff;
}
else {
    byteDx = byte(Dx*100);
    payload[6] = 0x00;
    payload[7] = byteDx & 0xff;
}
if (Dy < 0 ){
    byteDy = byte(Dy*(-100));
    payload[8] = 0xff;
    payload[9] = byteDy & 0xff;
}
else {
    byteDy = byte(Dy*100);
    payload[8] = 0x00;
    payload[9] = byteDy & 0xff;
}
if (Dz < 0 ){
    byteDz = byte(Dz*(-100));
    payload[10] = 0xff;
    payload[11] = byteDz & 0xff;
}
else{
    byteDz = byte(Dz*100);
    payload[10] = 0x00;
    payload[11] = byteDz & 0xff;
}
payload[12] = 0x00;
}

```

IV.3.3 Skematik rangkaian modul RF pada gateway

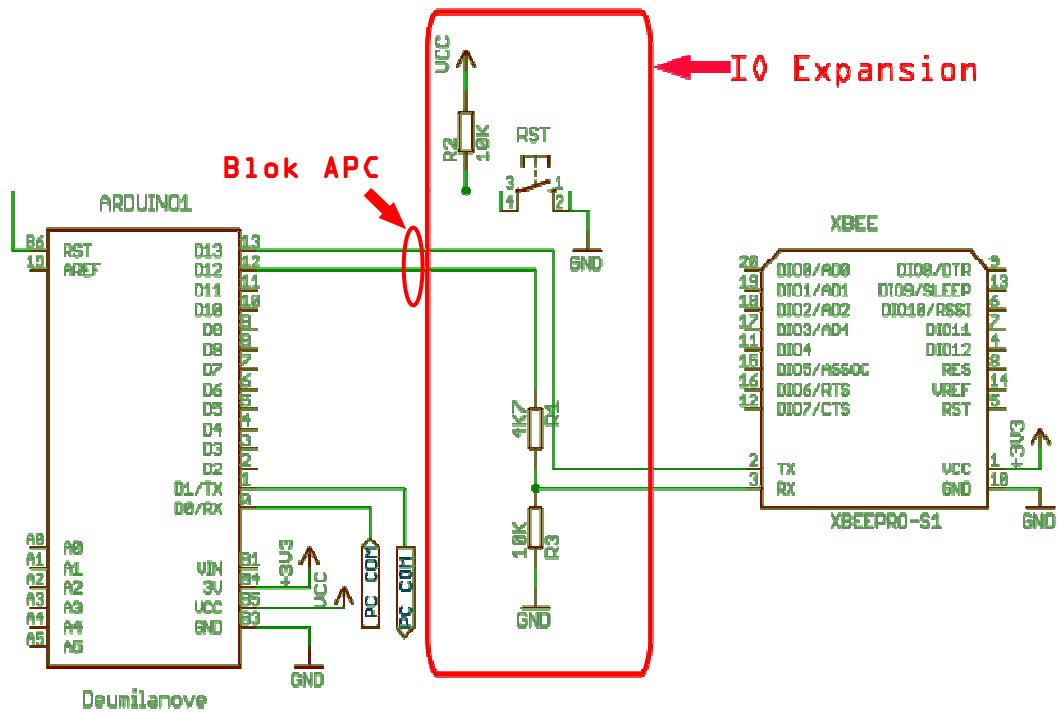
Fungsi utama dari *gateway* adalah untuk menerima data yang dikirim dari *router* serta mengirim data ke komputer server. Permasalahan yang dihadapi adalah perangkat modul RF dengan komputer server keduanya menggunakan mode komunikasi UART

dengan arduino deumilanove sedangkan jalur komunikasi UART pada arduino deumilanove hanya terdapat 1 buah yakni pin 0 dan pin 1 (pin RX dan pin TX).

Apabila menggunakan Xbee Shield seperti halnya *router* maka diperlukan pemindahan jumper JP1 dan JP2 (gambar 4.28) secara terus menerus sehingga jalur komunikasi UART pada arduino dapat digunakan secara bergantian dan transfer data baik dari modul RF ke arduino atau dari arduino ke komputer server dapat dilakukan. Hal ini sangat tidak efisien apabila dilakukan secara manual. Apabila menggunakan *switching* secara elektronis misalnya menggunakan transistor atau optocoupler maka pewaktuan proses pensaklaran tergantung dari karakteristik komponen yang digunakan sehingga dapat dimungkinkan adanya *error* pada proses transfer data.

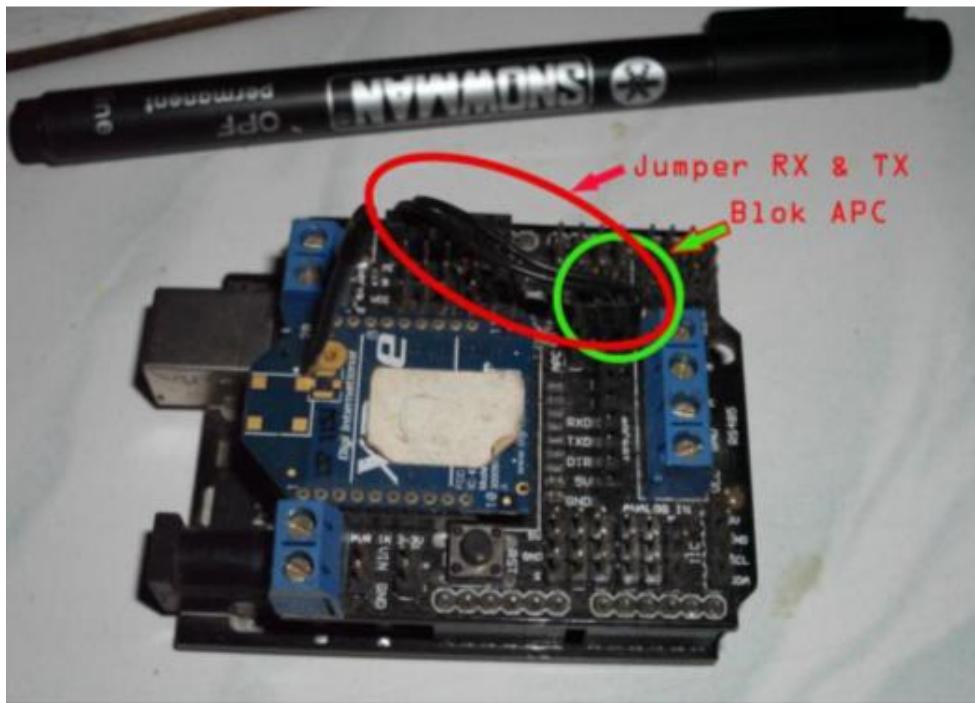
Salah satu solusi dari permasalahan tersebut adalah dengan menggunakan pendekatan *software* yakni dengan memanfaatkan *library* NewSoftSerial yang telah tersedia pada paket arduino IDE 0022. *Library* NewSoftSerial dibuat oleh David A. Mellis (2006). Dengan menggunakan *library* NewSoftSerial maka dapat membuat pin arduino yang bukan jalur komunikasi UART berfungsi sebagai jalur komunikasi UART sehingga dalam satu modul arduino deumilanove dapat dimungkinkan terdapat lebih dari satu jalur komunikasi UART.

Skematik rangkaian pada *gateway* sesuai dengan gambar 4.36 berikut.



Gambar 4.36 Skematik rangkaian pada gateway

Pada gateway, perantaraan Xbee dengan arduino dibantu menggunakan arduino IO Expansion. Posisi pin TX dan RX modul Xbee terhubung pada blok APC yang terlebih dahulu melalui penyesuaian level tegangan pada jalur RX Xbee. Untuk menghubungkan pin TX dan pin RX yang terdapat pada blok APC dari arduino IO Expansion dengan pin 12 dan pin 13 arduino deumilanove dilakukan dengan menambahkan kabel *jumper* seperti pada gambar 4.37.



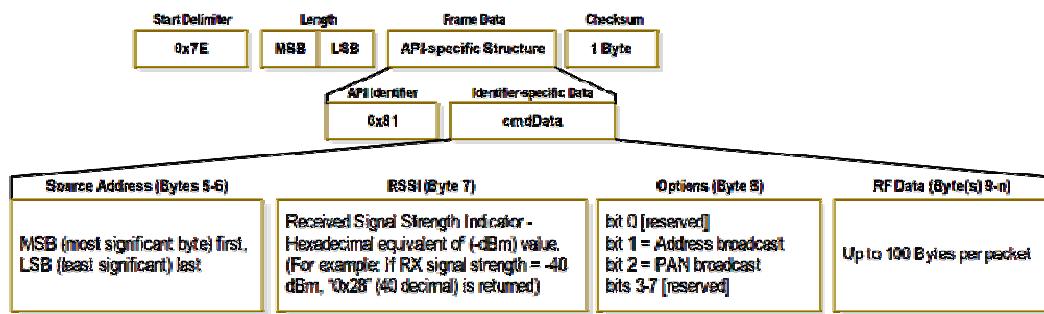
Gambar 4.37 Jumper RX dan TX pada blok APC dari arduino IO Expansion dengan pin 12 dan pin 13

IV.3.4 Algoritma pembacaan data pada gateway

Fungsi utama *gateway* adalah sebagai penerima data yang dikirim secara *wireless* oleh *router* dan mengirimkan kembali ke komputer server melalui komunikasi serial. Tugas ini sebenarnya dapat dilakukan dengan cara melewaskan langsung data yang diterima oleh Modul RF kemudian mengirimkan ke komputer server. Akan tetapi data yang diterima merupakan data yang ter-enkripsi dalam format API, sedangkan pengamatan terhadap data yang ter-enkripsi relatif lebih sulit jika dibandingkan dengan data yang tidak ter-enkripsi. Sehingga proses enkode frame data yang diterima dilakukan pada modul *gateway*.

Langkah pertama proses enkode frame data yang diterima adalah dengan mengenali tipe frame yang dikirim oleh *router*.

Pengalamatan modul yang digunakan adalah dengan menggunakan 16 bit alamat sehingga tipe frame API pada protokol zigbee dengan mode 16 bit alamat adalah 0x81 (Maxstream.Inc, 2007). Tipe frame data pada API *identifier* = 0x81 ditunjukan pada gambar 4.38 berikut.



Gambar 4.38 Tipe frame API untuk Xbee RX packet pada mode pengalamatan 16 bit (Maxstream.Inc, 2007)

Dapat dilihat format frame data pada tipe frame 0x81 bahwa dalam satu frame terdiri dari *start delimiter*(byte ke-1), data *length* (byte ke-2 dan ke-3), *Frame Data* (byte ke-4 hingga ke-n) dan *checksum* (byte terakhir).

Setelah mengetahui tipe frame API yang dikirim oleh router, langkah selanjutnya adalah menganalisa format frame data yang diterima. Frame data terdiri dari *Api Identifier* (byte ke-4) dan *identifier Specific Data* yang terdiri dari *Source Address* (byte ke-5 dan ke-6), *RSSI* (byte ke-7), *Options* (byte ke-8), serta *RF data* (byte ke-9 hingga ke-n). Didalam *RF Data* inilah data yang kita kirim berada.

Data yang kita kirim merupakan data yang disimpan dalam variabel *payload* pada modul *router* yang telah kita bahas sebelumnya.

Sehingga kita telah mengetahui jumlah *payload* yakni sebanyak 13 byte.

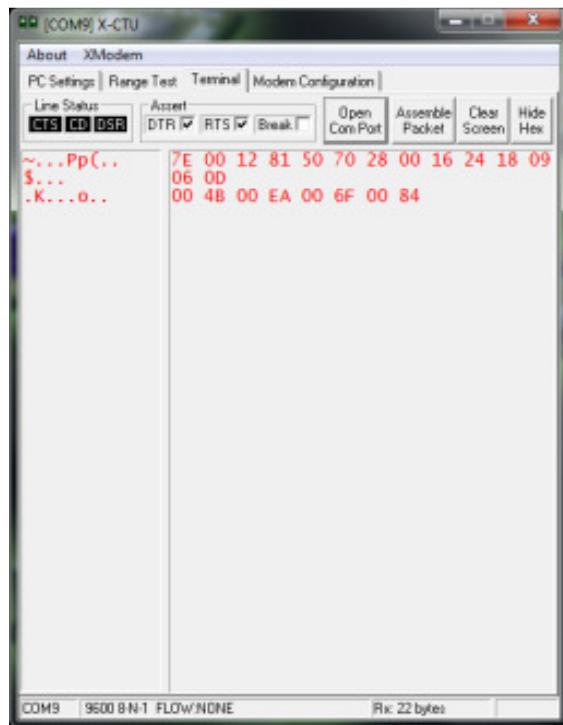
Sehingga data dari *payload* akan menempati byte ke-9 hingga ke-21.

Sampai disini kita dapat menentukan format frame data yang akan diterima oleh *gateway* yaitu sebagai berikut.

byte ke-1	=	0x7E
byte ke-2	=	0x00
byte ke-3	=	0x12
byte ke-4	=	0x81
byte ke-5	=	MSB address
byte ke-6	=	LSB address
byte ke-7	=	RSSI
byte ke-8	=	0x00
byte ke-9	=	jam
byte ke-10	=	menit
byte ke-11	=	detik
byte ke-12	=	tanggal
byte ke-13	=	bulan
byte ke-14	=	tahun
byte ke-15	=	flag Dx
byte ke-16	=	Dx
byte ke-17	=	flag Dy
byte ke-18	=	Dy
byte ke-19	=	Flag Dz
byte ke-20	=	Dz
byte ke-21	=	0x00
byte ke-22	=	checksum

Langkah selanjutnya adalah melakukan uji coba membaca data yang diterima oleh modul Xbee, dengan cara menghubungkan modul Xbee ke komputer server kemudian mengamati data yang diterima pada *softwere* terminal. Untuk uji coba ini dapat memanfaatkan X-CTU , CoolTerm atau softwere terminal yang lain yang dapat menampilkan bilangan heksadesimal dari *byte* data yang diterima.

Hasil pengujian menggunakan X-CTU dapat dilihat pada gambar 4.39 dibawah.



Gambar 4.39 Data yang diterima oleh modul Xbee

Dari data yang diterima oleh modul Xbee yang ditunjukan pada gambar 4.33 dapat dianalisa sebagai berikut.

Byte ke-1 bernilai 0x7E yang berarti *start delimiter*. Kemudian dilanjutkan dengan byte ke-2 dan ke-3 secara berturut – turut merupakan MSB dan LSB dari *Data Length* yaitu jumlah byte dari *API Identifier* hingga byte terakhir sebelum *checksum* atau dari byte ke-4 hingga byte ke-21 yang berjumlah 17 byte. (dalam format heksadesimal = 0x12).

Modul Xbee yang digunakan pada router terdiri dari 2 buah yang masing – masing memiliki alamat 5070 dan 4D36 sehingga jika

byte ke-5 digeser ke kiri 8 bit kemudian ditambahkan dengan *byte* ke-6 akan memiliki dua kemungkinan nilai yaitu 0x5070 atau 0x4D36. Nilai ini tergantung dari modul mana data ini dikirimkan. Karena data menunjukan 0x5070 maka modul Xbee pengirim mempunyai alamat 0x5070.

Byte ke-7 bernilai 0x28, sedangkan *byte* ini merupakan nilai RSSI negatif dalam skala dBm (desibel miliwat). Sehingga pada pengujian diatas menunjukan bahwa RSSI bernilai -40 dBm.

Byte ke-8 bernilai 0x00 menunjukan bahwa opsi pengiriman yang digunakan oleh pengirim secara *reserved* yang artinya data yang dikirim hanya untuk alamat tertentu saja. Jika *byte* ini bernilai 0x01 maka pengiriman yang digunakan secara broadcast dalam satu PAN (*Personal Area Network*) sedangkan jika bernilai 0x02 maka opsi pengiriman yang digunakan oleh pengirim adalah secara broadcast antar PAN.

Byte ke-9 hingga 14 merupakan data waktu pengiriman data yang didapat dari unit RTC pada modul *router*. Secara berturut – turut data ini menujukan waktu pukul 22:36 .24 tanggal 9/6/13.

Byte ke-15, ke-17, dan ke-19 merupakan *flag* atau penanda nilai bilangan pada *byte* selanjutnya yang telah diset sebelumnya pada program pengiriman data pada *router*. Sehingga dari data yang diterima didapat informasi bahwa *byte* ke-16, ke-18 dan ke-20 semuanya bernilai positif.

Byte ke-16, ke-18 dan ke-20 secara berturut – turut merupakan data perpindahan pada sumbu X, Sumbu Y dan sumbu Z dan nilainya menunjukan perpindahan sebesar Dx = 75 cm, Dy = 234 cm dan Dz = 111 cm.

Byte ke-21 merupakan *byte* yang didapat dari *payload[13]* pada program *router* yang berfungsi untuk menghindari adanya *escape byte*. Sedangkan *byte* ke-22 adalah *byte* terakhir atau *checksum*. *Byte* ini dapat digunakan sebagai pengkoreksi data yang dikirim benar atau tidak. Untuk menguji jika data dikirim dengan sempurna dilakukan dengan cara menjumlahkan seluruh *byte* yang diterima kecuali *byte* ke-1 sampai *byte* ke-3. Data yang dikirim tidak cacat jika hasil penjumlahan pada 1 *byte* LSB bernilai 0xff. Jadi hasil verifikasi dari data yang diperoleh diatas adalah sebagai berikut.

$$\begin{aligned} 0x81 + 0x50 + 0x70 + 0x28 + 0x00 + 0x16 + 0x24 + 0x18 \\ + 0x06 + 0x0D + 0x00 + 0x4B + 0x00 + 0xEA + 0x00 + 0x6F \\ + 0x00 + 0x84 = 0x3FF \end{aligned}$$

Nilai 1 *byte* LSB dari hasil perhitungan diatas adalah 0xFF sehingga dapat dipastikan bahwa data yang dikirim tidak ada cacat.

Langkah selanjutnya adalah menyusun algoritma yang akan digunakan untuk memprogram mikrokontroler. Penyusunan algoritma program pada *gateway* didasarkan pada 2 pokok permasalahan yaitu :

1. Jalur komunikasi.
2. Informasi apa saja yang akan dikirim ke komputer server.

Penentuan jalur komunikasi telah dibahas sebelumnya yaitu dengan pendekatan software dengan memanfaatkan *library* NewSoftSerial sehingga diawal program perlu dimasukan header *NewSoftSerial.h*. Setelah memasukan *header* kemudian membuat *object* dengan *class* NewSoftSerial dan menentukan pin Rx dan Tx yang digunakan untuk komunikasi UART. *Class* NewSoftSerial memiliki beberapa *public methods* berikut.

```
NewSoftSerial(uint8_t receivePin, uint8_t transmitPin,  
    bool inverse_logic = false);  
~NewSoftSerial();  
void begin(long speed);  
void end();  
int read();  
uint8_t available(void);  
bool active()  
bool overflow()  
static int library_version()  
static void enable_timer0(bool enable);  
void flush();
```

Jalur komunikasi yang terbentuk akan digunakan untuk transfer data antara modul Xbee dengan mikrokontroler sehingga *object* ini diberi nama xbee dengan pin Tx pada pin 12 dan pin Rx pada pin 13. Sampai disini kita telah memiliki jalur komunikasi UART pada pin 12 dan pin 13.

Potongan listing program untuk konstruktor jalur UART ini sebagai berikut.

```
#include <NewSoftSerial.h>  
NewSoftSerial xbee (13,12);
```

Selanjutnya adalah menyiapkan variabel untuk menampung *byte* data yang diterima dari modul Xbee. Mengingat bahwa didalam 1 frame

terdapat 22 *byte* data sehingga variabel ini berbentuk *array* dari `uint8_t` dengan lebar *array* 22.

Langkah selanjutnya adalah mengisi *array* dengan *byte* data yang diterima dari modul Xbee dan kemudian memilih data mana saja yang akan dikirim ke komputer server. Proses ini dilakukan dengan membaca *byte* data satu persatu dan memasukanya kedalam tiap – tiap indeks *array*.

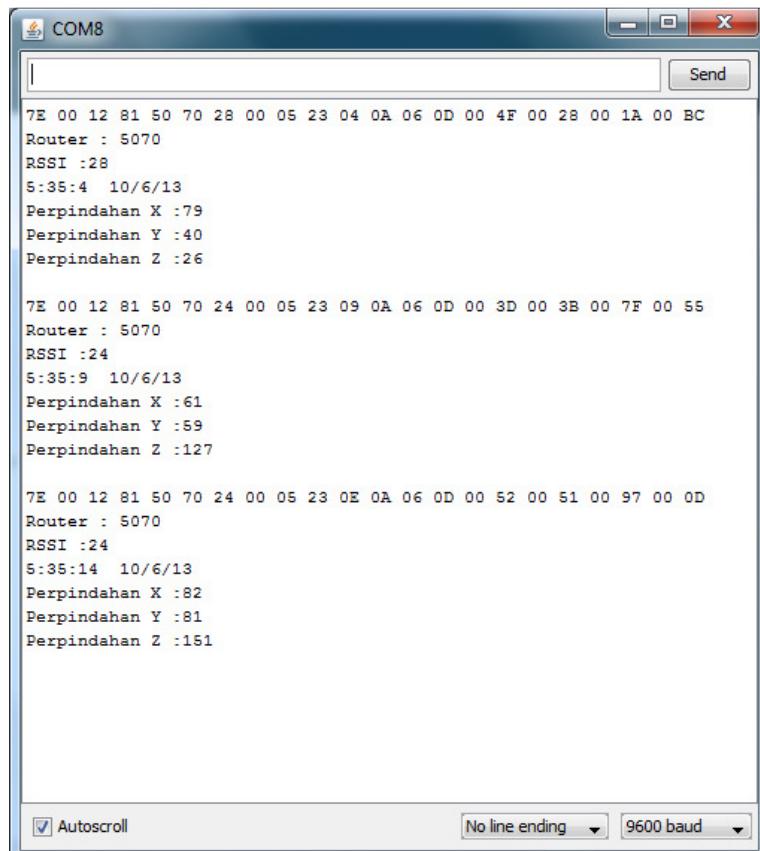
Informasi yang perlu ditampilkan di komputer server diantaranya adalah alamat pengirim data, data RSSI, data waktu pengiriman, serta data perpindahan sehingga *array* yang akan diproses nilainya pada indeks 5,6,7,9 hingga 20. Mengingat kembali bahwa data akselerometer memiliki *flag* pada *byte* ke-15, 17 dan 19 maka untuk menentukan tanda bilangan pada data percepatan dilakukan dengan memanfaatkan instruksi `if (conditional)` dengan parameter yang diuji adalah array yang menunjukkan *flag* untuk data perpindahan yakni pada indeks ke-14, 16 dan 18 (indek *array* pertama adalah 0). Potongan program pada algoritma utama penerimaan data dari modul Xbee dan mengirimkan data ke komputer server adalah sebagai berikut.

```

void loop() {
    int sX, sY, sZ;
    while (xbee.available() > 0) {
        int b = readByte();
        if (b == 0x7E) {
            packet[0] = b;
            packet[1] = readByte();
            packet[2] = readByte();
            int Length = ((packet[1] << 8) | packet[2]);
            for(int i=1;i<=(Length+1);i++) {
                packet[2+i] = readByte();
            }
            printPacket(Length+4);
            int frametype = packet[3];
            if (frametype == 0x81) {
                int router = (packet[4] << 8) | packet[5];
                Serial.print("Router : ");
                Serial.println(router,HEX);
                int RSSI = packet[6];
                Serial.print("RSSI :");
                Serial.println(RSSI, HEX);
                int hour = packet[8];
                int minute = packet[9];
                int second = packet[10];
                int dayOfMonth = packet[11];
                int month = packet[12];
                int year = packet[13];
                Serial.print(hour,DEC);           Serial.print(":");
                Serial.print(minute,DEC);         Serial.print(":");
                Serial.print(second,DEC);         Serial.print(" ");
                Serial.print(dayOfMonth,DEC);     Serial.print("/");
                Serial.print(month,DEC);          Serial.print("/");
                Serial.println(year,DEC);
                if (packet[14] == 0x00){
                    sX = packet[15] ;
                }
                if (packet[14] == 0xff){
                    sX = packet [15]*(-1);}
                if (packet[16] == 0x00){
                    sY = packet[17] ;
                }
                if (packet[16] == 0xff){
                    sY = packet [17]*(-1);}
                if (packet[18] == 0x00){
                    sZ = packet[19] ;
                }
                if (packet[18] == 0xff){
                    sZ = packet [19]*(-1);}
                Serial.print("Perpindahan X :");
                Serial.println(sX,DEC);
                Serial.print("Perpindahan Y :");
                Serial.println(sY,DEC);
                Serial.print("Perpindahan Z :");
                Serial.println(sZ,DEC);
            }
        }
    }
}

```

Hasil yang didapat pada program pembacaan dan pengiriman data pada *gateway* ditunjukan oleh gambar 4.40 dibawah ini.



The screenshot shows a terminal window titled "COM8". The window contains three sets of sensor data, each starting with a hex string and followed by descriptive text. The data is as follows:

```
7E 00 12 81 50 70 28 00 05 23 04 0A 06 0D 00 4F 00 28 00 1A 00 BC
Router : 5070
RSSI :28
5:35:4 10/6/13
Perpindahan X :79
Perpindahan Y :40
Perpindahan Z :26

7E 00 12 81 50 70 24 00 05 23 09 0A 06 0D 00 3D 00 3B 00 7F 00 55
Router : 5070
RSSI :24
5:35:9 10/6/13
Perpindahan X :61
Perpindahan Y :59
Perpindahan Z :127

7E 00 12 81 50 70 24 00 05 23 0E 0A 06 0D 00 52 00 51 00 97 00 0D
Router : 5070
RSSI :24
5:35:14 10/6/13
Perpindahan X :82
Perpindahan Y :81
Perpindahan Z :151
```

At the bottom of the window, there are three buttons: "Autoscroll" (checked), "No line ending", and "9600 baud".

Gambar 4.40 Hasil runing program pada gateway

IV.4 Pengujian RSSI (*Receive Signal Strength Indicator*)

Untuk pengujian RSSI dilakukan dengan 2 cara yakni menguji kekuatan sinyal pada daerah yang tidak ada halangan atau LOS (*Line Of Sight*) dan menguji kekuatan sinyal pada daerah yang terdapat halangan. Skematik yang digunakan untuk pengujian ini adalah sebagai berikut (gambar 4.41)



Gambar 4.41 Skematik rangkaian pengujian RSSI

Perangkat yang dibutuhkan untuk pengujian ini adalah sebagai berikut.

- Notebook sebagai media penampil dan penyimpan data.
- Modul penerima terdiri dari modul Xbee Pro Series 1 dan Arduino.
- Modul pengirim terdiri dari modul Xbee Pro Series 1, Arduino dan batere *lithium* 7,4 Volt sebagai catu daya modul.
- Meteran gulung
- Tali/benang dengan panjang 20 meter.
- Tiang penyangga modul dengan panjang 1,3 meter hingga 1,5 meter.

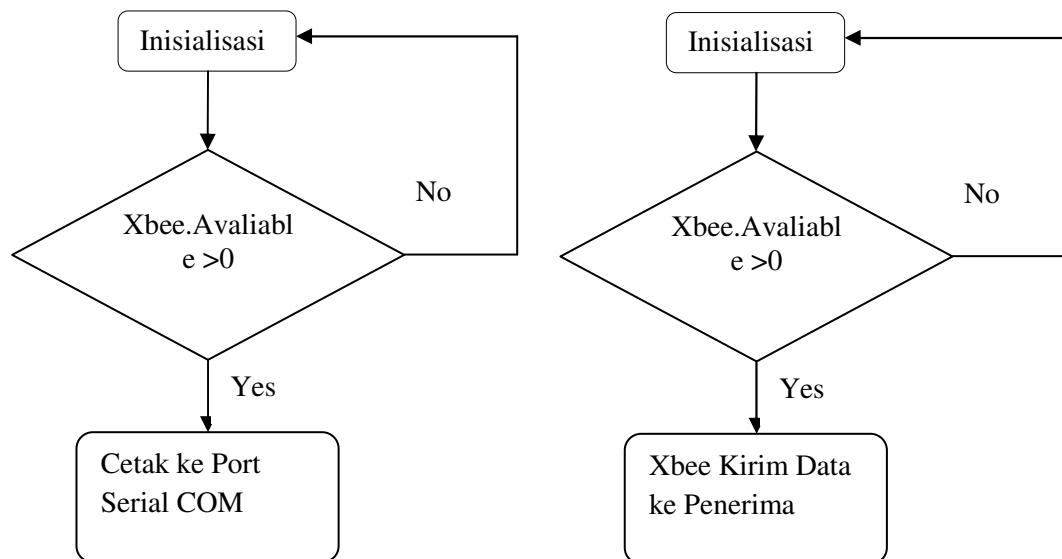
Untuk seting parameter dari tiap modul Xbee Pro yang digunakan sesuai dengan tabel 4.15 berikut.

Tabel 4.15 Seting parameter Xbee untuk pengujian RSSI

Parameter	Pengirim	Penerima
BD (<i>Baud Rate</i>)	3 (9600 bps)	3 (9600 bps)
DL (<i>Destination Address Low</i>)	7E63	FFFF
MY (<i>I6 bit Source Address</i>)	4D36	7E63
PL (<i>Power Level</i>)	4 (10 dBm)	4 (10 dBm)

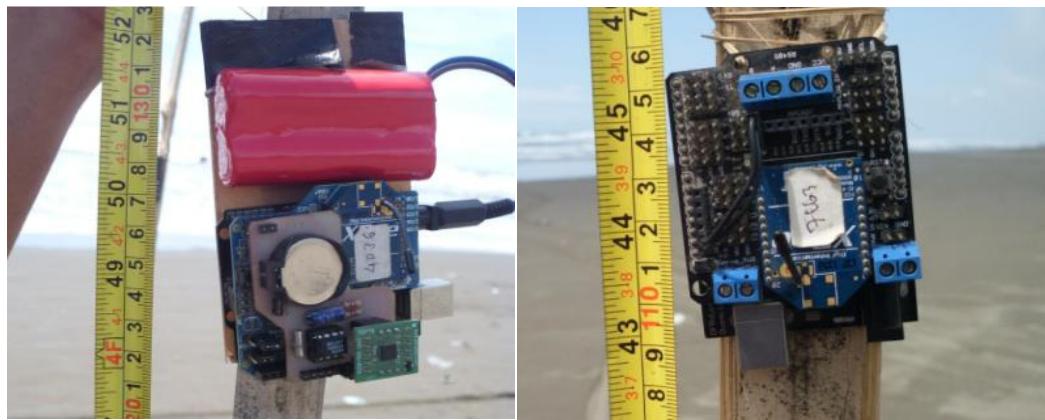
Untuk pengujian RSSI ini berkaitan erat dengan *Power Level* dari modul yang digunakan. Pada parameter PL (Power Level) diseting 4 dimana modul yang digunakan adalah Xbee Pro yang varian internasional sehingga nilai 4 berarti *power level* dari modul diseting -10 dBm.

Algoritma yang digunakan untuk pengujian RSSI ini adalah sebagai berikut (gambar 4.42).



Gambar 4.42 Flowchart algoritma modul penerima (kiri) dan modul pengirim (kanan) yang digunakan pada pengujian RSSI

Untuk pengujian LOS dilakukan di pantai Widarapayung kecamatan Binangun kabupaten Cilacap. Pengujian dilakukan hingga jarak 500 meter dengan data yang diambil per 20 meter. Untuk menghindari adanya halangan sehingga antara modul pengirim dan penerima benar – benar LOS, tiap modul diletakan pada ketinggian 1 meter hingga 1,3 meter diatas tanah (gambar 4.43).



Gambar 4.43 Tinggi modul dari permukaan tanah (kiri : Pengirim, kanan : Penerima)

Data yang diperoleh dari pengujian ini adalah sebagai berikut (Tabel 4.16).

Tabel 4.16 Data pengujian RSSI modul terhubung secara LOS

NO	Jarak (m)	RSSI Sample ke -n (dalam Heksadesimal)										RSSI Rata - Rata
		1	2	3	4	5	6	7	8	9	10	
1	20	3C	3C	3C	3C	3C	3C	3C	3C	3C	3C	-60 dBm
2	40	49	4C	4D	4D	4D	4D	4D	4E	4A	4C	-76 dBm
3	60	4F	4D	4F	4F	50	50	4F	55	53	4D	-79 dBm
4	80	54	57	58	62	5F	5D	59	58	62	53	-90 dBm
5	100	5D	59	62	5A	59	5F	62	65	57	5D	-93 dBm
6	120	62	5F	62	62	5D	5F	5D	5F	65	5F	-96 dBm
7	140	56	5A	62	58	62	5A	65	5D	62	65	-94 dBm
8	160	65	62	62	62	62	62	65	5F	65	65	-98 dBm
9	180	5D	65	62	62	62	65	62	62	5C	62	-97 dBm
10	200	5D	62	65	65	62	5F	62	65	62	62	-98 dBm
11	220	62	65	65	65	65	65	65	65	65	64	-100 dBm
12	240	62	59	54	56	54	54	55	5C	58	54	-87 dBm
13	260	65	5C	58	58	59	5A	5C	5C	5A	5A	-91 dBm
14	280	53	57	52	57	54	52	51	50	4F	52	-82 dBm
15	300	59	59	57	57	57	57	57	54	54	56	-86 dBm
16	320	5D	5A	5C	5A	5A	5D	5F	5C	5D	62	-92 dBm
17	340	55	53	50	56	58	5A	62	5C	5F	5D	-89 dBm
18	360	62	5D	62	5F	5F	5D	5D	5F	62	5F	-95 dBm
19	380	65	5D	5F	62	62	5C	5D	5C	5D	5C	-94 dBm
20	400	62	62	5D	65	65	65	65	65	65	63	-99 dBm

21	420	65	65	62	62	65	5D	62	62	62	65	62	-98 dBm
22	440	5F	5C	62	62	62	5F	5F	62	5D	5F	5F	-95 dBm
23	460	62	62	5F	62	65	65	65	62	62	65	62	-98 dBm
24	480	65	62	62	62	5F	62	62	62	55	62	60	-96 dBm
25	500	65	65	65	65	62	62	62	65	62	65	63	-99 dBm

Pada pengujian LOS, RSSI terendah yang diterima adalah sebesar -100 dBm. Pada jarak 500 meter data masih dapat diterima dengan baik sehingga besar kemungkinan pada jarak lebih dari 500 meter masih dapat melakukan komunikasi karena pada *data sheet* Xbee Pro Series 1 jangkauan maksimal untuk kondisi LOS adalah sekitar 1.5 Km.

Modul Xbee yang digunakan adalah Modul Xbee Pro Series 1 dengan tipe antena *monopole (integrated whip)*. Berdasarkan *data sheet* Xbee Pro Series 1 (Maxstream, 2007) *gain* antena adalah 1.5 dBi sehingga dari data yang ada dapat dibuat grafik *pathloss* terhadap jarak dengan perhitungan sebagai berikut.

$$RSSI = Pt + Gt + Gr - Pl \quad \dots \dots \dots \quad (4.15)$$

Dimana :

RSSI = Daya diterima (-dBm)

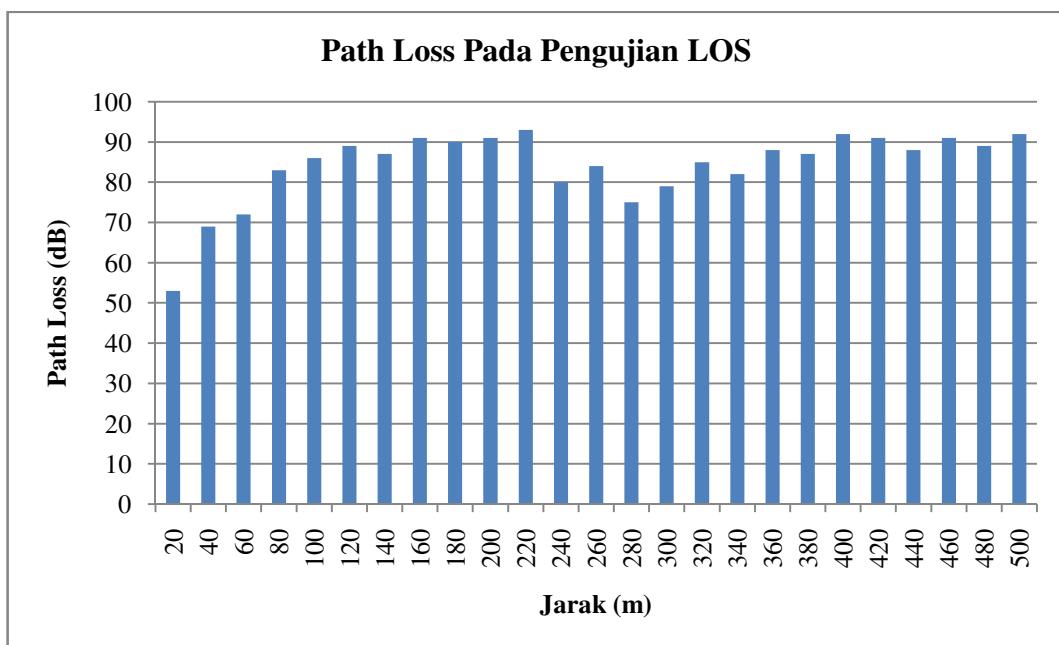
Pt = Daya pancar / *Power level* (-dBm)

Gt = Gain antena pengirim (dBi)

Gr = Gain antena penerima (dBi)

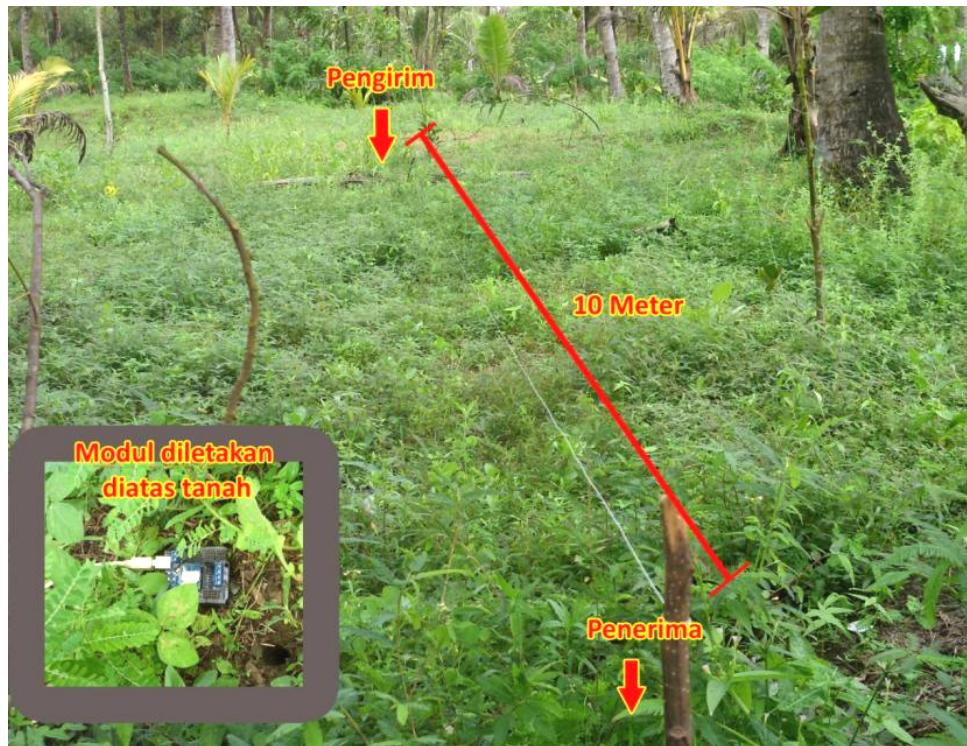
Pl = *Path loss* total (dB)

Dan hasilnya adalah sebagai berikut.(gambar 4.44)



Gambar 4.44 Grafik perbandingan nilai path loss terhadap jaraks

Pengujian RSSI berikutnya adalah pengujian modul pada daerah berpenghalang (modul terinterferensi). Pengujian ini dilakukan di pesisir Pantai Widara Payung dimana daerah tersebut merupakan daerah perkebunan yang ditanami pohon kelapa dan dibawahnya ditanami palawija serta ada beberapa yang tidak ditanami palawija sehingga ditumbuhi tumbuhan liar dan perdu. Gambar 4.45 menunjukan lokasi pengujian RSSI untuk daerah yang berpenghalang.



Gambar 4.45 Lokasi pengujian nilai RSSI pada medan berpenghalang

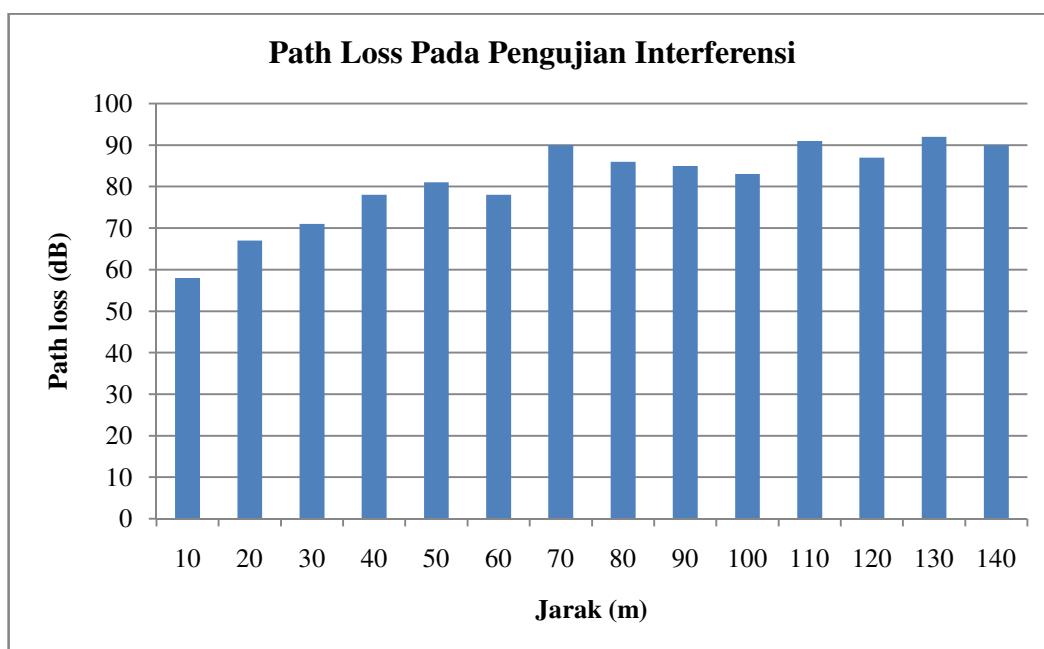
Data RSSI yang diambil tiap 10 meter dan dilakukan hingga tidak ada data yang terkirim. Dari hasil percobaan, jarak terjauh data dapat diterima oleh modul penerima adalah 140 meter dan RSSI terendah adalah -99 dBm.

Data hasil pengujian ditunjukkan oleh tabel 4.17 dan grafik perbandingan nilai RSSI terhadap jarak ditunjukkan oleh gambar 4.46.

Tabel 4.17 Data hasil pengujian RSSI pada medan berpenghalang

NO	Jarak (m)	RSSI Sample ke -n (dalam Heksadesimal)										RSSI Rata - Rata	
		1	2	3	4	5	6	7	8	9	10		
1	10	42	42	42	3F	3F	40	3C	43	42	47	41	-65 dBm
2	20	4F	49	4C	4E	46	4E	47	46	44	4F	4A	-74 dBm
3	30	4F	50	50	58	4A	49	4D	51	49	4E	4E	-78 dBm
4	40	58	56	53	52	52	59	53	52	5F	53	55	-85 dBm
5	50	54	5F	57	5F	56	54	4D	62	59	57	58	-88 dBm
6	60	4F	58	59	56	59	4C	5A	52	53	5F	55	-85 dBm

7	70	62	5D	65	65	62	59	65	65	65	59	61	-97	dBm
8	80	59	5D	5A	5F	59	5D	5C	65	5F	65	5D	-93	dBm
9	90	5D	5D	59	5D	5A	62	58	65	57	5A	5C	-92	dBm
10	100	5A	5F	58	5D	5A	5A	5C	5C	5D	53	5A	-90	dBm
11	110	65	5F	65	65	5F	65	62	62	5F	62	62	-98	dBm
12	120	62	5D	65	5F	58	5C	5D	62	62	5D	5E	-94	dBm
13	130	65	65	5F	65	62	65	65	62	62	65	63	-99	dBm
14	140	65	5F	5F								61	-97	dBm



Gambar 4.46 Grafik perbandingan nilai RSSI terhadap jarak pada medan berpenghalang (modul terinterferensi)

Agung Budiaji (2012) dari Fakultas Sains dan Teknik Universitas Jenderal Soedirman Purwokerto melakukan pengujian RSSI pada modul Xbee Series 1 untuk kondisi LOS dan terinterferensi. Jangkauan maksimal pada Xbee Series 1 untuk kondisi LOS adalah 90 meter dan untuk kondisi terinterferensi adalah 80 meter. RSSI terendah untuk kedua kondisi adalah -85 dBm.

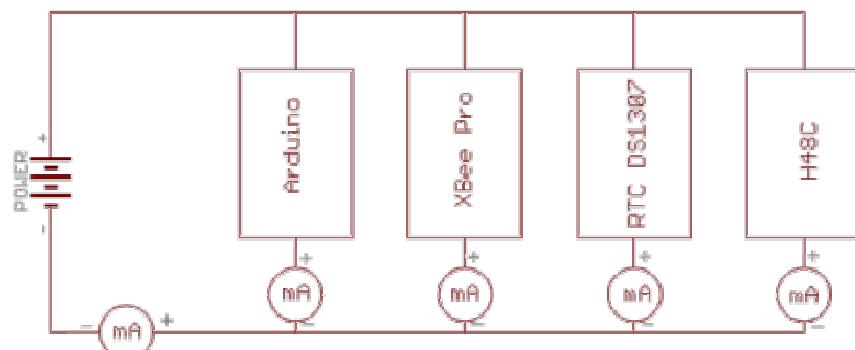
Dibandingkan dengan penelitian sebelumnya (Budiaji, Agung.,2012) penggunaan Xbee Pro Series 1 memiliki keunggulan dalam hal jangkauan maksimal dan RSSI terendah yang dapat diterima.

IV.5 Konsumsi Daya

Salah satu aspek terpenting dalam jaringan sensor nirkabel adalah pengetahuan mengenai konsumsi daya dari modul yang digunakan. Hal ini dikarenakan *node* akan diletakan di daerah tertentu dan terkadang jauh jala – jala PLN. Sehingga informasi konsumsi daya penting untuk menentukan tipe catu daya yang akan digunakan khususnya konsumsi daya *router*.

Dalam sebuah *node router* terdiri dari 4 bagian utama yaitu kontroler, unit RF, unit RTC dan Sensor. Sehingga untuk mengetahui konsumsi daya sebuah *router* diperlukan pengukuran pada masing – masing bagian.

Pengukuran daya dapat dilakukan dengan mengukur konsumsi arus modul. Skematik pengukuran arus dari sebuah *router* ditunjukkan oleh gambar 4.47 dibawah.



Gambar 4.47 Skematik pengukuran daya dari sebuah *router*

Pada pengukuran digunakan catu daya berupa 2 buah sel baterai *Lithium Polimer* dengan tegangan masing – masing 3.6 V dengan kapasitas 2600 mAH dan disusun seri sehingga tegangan catu daya keseluruhan adalah 7.2 V.

Pengukuran dilakukan pada modul yang sedang menjalankan program *router* dan mengirimkan data melalui modul RF ke *gateway* dengan interval pengiriman data 5 detik.

Hasil yang didapat dari pengukuran ditunjukan pada tabel 4.18 dibawah.

Tabel 4.18 Hasil pengukuran konsumsi arus modul *router*

Unit	Arus
Kontroler (Arduino)	20 mA
RF (Xbee Pro)	62 mA
RTC (DS1307)	2 mA
Akselerometer (H48C)	1 mA
Arus Total	85 mA

Dengan arus total 85 mA dan tegangan kerja adalah 7.2 V sehingga konsumsi daya modul adalah 612 mW. Dengan kapasitas baterai 2600 mAH idealnya baterai dapat mensuplai modul selama 30.5 jam akan tetapi pada kenyataanya dapat bervariasi.

Jika menggunakan baterai sebagai catu daya modul, kapasitas baterai sangat menentukan lamanya waktu kerja modul. Sehingga untuk membuat modul bekerja lebih lama dapat dilakukan dengan beberapa cara seperti menambah kapasitas baterai, meminimalisir kerja kontroler, serta memperpanjang jeda waktu pengiriman data.

Menambah kapasitas baterai dapat dilakukan dengan memilih baterai yang kapasitasnya tinggi atau menggabungkan beberapa sel baterai secara paralel. Akan tetapi baterai memiliki hambatan dalam yang menyebabkan baterai mengalami *self-discharge* meskipun baterai tidak digunakan.

Semakin tinggi kerja kontroler menyebabkan kontroler mengkonsumsi arus lebih banyak sehingga untuk memperkecil konsumsi arus dapat dilakukan dengan membuat kerja kontroler seringan mungkin. Ini dapat dilakukan dengan beberapa cara yaitu membuat algoritma yang lebih ringkas, menghindari adanya operasi matematikal yang kompleks, mengganti operasi pembagian dengan operasi perkalian atau dengan operasi penggeseran bit jika mungkin, mengurangi penggunaan tipe *float* karena tipe ini membutuhkan alokasi 32 bit untuk *single precision* dan 64 bit untuk *double precision* sehingga melibatkan operasi bilangan pada *float* akan lebih lama dari pada operasi pada bilangan bulat (*integer*).

Selang waktu pengiriman data berpengaruh sangat besar pada konsumsi daya dari sebuah modul karena modul RF membutuhkan arus yang tinggi pada saat mengirim data. XbeePro sendiri membutuhkan arus 250 mA untuk melakukan pengiriman data dan hanya 55 mA pada saat *idle*. Sehingga semakin cepat interval pengiriman data akan berakibat pada semakin besarnya konsumsi daya dari modul.

BAB V

PENUTUP

IV.1 Kesimpulan

Akselerometer dapat digunakan untuk melakukan pengukuran data perpindahan sehingga dapat diaplikasikan untuk melakukan pengukuran pergerakan material tanah. Data dari akselerometer masih terdapat banyak *noise* sehingga diperlukan pemfilteran data.

Setelah menggunakan filter *Bessel* orde 10, akurasi rata - rata pembacaan akselerometer pada pergerakan sumbu x adalah 93.7 %, pada pergerakan sumbu y adalah 95.5 %, dan pada sumbu z adalah 92.6 %.

Pada pengujian komunikasi data pada kondisi LOS, modul RF masih dapat melakukan komunikasi pada jarak 500 meter hanya saja nilai RSSI nya sudah melemah hingga -97 dBm. Sedangkan pada pengujian RSSI pada daerah terinterferensi, jangkauan maksimal komunikasi dapat dilakukan adalah sebesar 140 meter dengan nilai RSSI -97 dBm.

IV.2 Saran

Penggunaan akselerometer untuk pengukuran percepatan dinamis masih terdapat nilai percepatan yang dipengaruhi oleh percepatan gravitasi sehingga diperlukan tambahan sensor untuk memberikan koreksi kemiringan sensor, ini dapat dilakukan dengan menambahkan *gyro*.

DAFTAR PUSTAKA

A.E Putra. 2009. *Embeded System Menggunakan Mikrokonroler dan Pemrograman C.* Andi : Yogyakarta

ATMEL Corp. 2003. *Atmega8 And Atmega8L Preliminary.* ATMEL Corporation : Parkay

BNPB^a. 2012. *Rekapitulasi Kejadian Bencana Tahun 2011 di Indonesia .* <http://www.bnrb.go.id/website/asp/benc.asp?p=11> diakses pada: Kamis 29 Maret 2011.

BNPB^b. 2007. *Pengenalan Karakteristik Bencana dan Upaya Mitigasinya di Indonesia edisi III.* Direktorat Mitigasi : Jakarta

Budiaji, Agung. 2012. *Analisis Quality Of Servis (QOS) Pada Jaringan Sensor Nirkabel Menggunakan Modul Komunikasi Xbee Series 1.* Universitas Jenderal Soedirman : Purwokerto

Digi Manual. 2008. *Xbee Znet 2.5 / Xbee PRO Znet 2.5 OEM RF Modules.* Digi International Inc : United States

Digi Manual. 2009. *Xbee/Xbee-PRO RF Modules Product Manual v1.xEx – 802.15.4 Protocol.* Digi International Inc : United States

Ergen , S.C. 2004. *ZigBee / IEEE802.15.4 Summary .* <http://www.sinemerger.com/academic/publications/zigbee.pdf> diakses tanggal 8 November 2011.

- Faludi, Robert. 2010. *Buliding Wireless Sensor Network*. O'Reilly : United States
- Hill, Jasson Lester. 2003. *System Architecture Of Wireless Sensor Network*. University Of California : Berkeley
- I.F Akyidiz dan Mehmet Can Vuran. 2010. *Wireless Sensor Network*. Wiley: United Kingdom

LAMPIRAN - LAMPIRAN