



INSTITUT TEKNOLOGI BANDUNG

PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 **Telepon** (022)2508135-36, **Faks** (022)250 0940
BANDUNG 40132

Dokumentasi Produk Tugas Akhir

Lembar Sampul Dokumen

Judul Dokumen

TUGAS AKHIR TEKNIK ELEKTRO:
*Sistem Jaringan Detektor Gempa dan Tsunami
Decision Support System*

Jenis Dokumen

PROPOSAL

Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB

Nomor Dokumen

B100-01-TA1617.01.069

Nomor Revisi

02

Nama File

B100-01-TA1617.01.069.docx

Tanggal Penerbitan

12 May 2017

Unit Penerbit

Prodi Teknik Elektro – ITB

Jumlah Halaman

22

(termasuk lembar sampul ini)

Data Pengusul

Pengusul	Nama	Christoporus Deo Putratama	Jabatan	Anggota Kelompok
	Tanggal	9 September 2016	Tanda Tangan	
	Nama	Bramantio Yuwono	Jabatan	Anggota Kelompok
	Tanggal	9 September 2016	Tanda Tangan	
	Nama	Kevin Shidqi Prakoso	Jabatan	Anggota Kelompok
	Tanggal	9 September 2016	Tanda Tangan	
Pembimbing	Nama	Dr. techn. Ary Setijadi Prihatmanto	Jabatan	Dosen Pembimbing
	Tanggal	9 September 2016	Tanda Tangan	

Lembaga

Program Studi Teknik Elektro

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Alamat

Labtek V, Lantai 2, Jalan Ganesha no. 10, Bandung

Telepon : +62 22 250 2260 Faks :+62 22 253 4222

Email:stei@stei.itb.ac.id

Nomor Dokumen: B100-01-TA1617.01.069 Nomor Revisi: 02 Tanggal: 12/05/2017 Halaman 1 dari 22

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas ijin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

DAFTAR ISI

DAFTAR ISI.....	2
CATATAN SEJARAH PERBAIKAN DOKUMEN.....	3
PROPOSAL PROYEK PENGEMBANGAN SISTEM JARINGAN DETEKTOR GEMPA DAN TSUNAMI <i>DECISION SUPPORT SYSTEM</i>	4
1 PENGANTAR	4
1.1 RINGKASAN ISI DOKUMEN	4
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN	4
1.3 REFERENSI	4
1.4 DAFTAR SINGKATAN.....	5
2 DEVELOPMENT PROJECT PROPOSAL.....	6
2.1 MASALAH DAN TUJUAN	6
2.2 ANALISIS UMUM.....	9
2.3 PRODUCT CHARACTERISTIC	10
2.3.1 <i>Fitur Utama</i>	10
2.3.2 <i>Fitur Dasar</i>	10
2.3.3 <i>Fitur Tambahan</i>	11
2.4 COST ESTIMATE	12
2.5 ANALISIS FINANSIAL.....	15
2.6 SKENARIO PEMANFAATAN PRODUK	18
2.7 SKENARIO PENGEMBANGAN PRODUKSI DAN PEMASARAN	18
2.8 KESIMPULAN DAN RINGKASAN	19
3 LAMPIRAN.....	20
3.1 DATA ANGGOTA TIM	20

Catatan Sejarah Perbaikan Dokumen

VERSI	TANGGAL	OLEH	PERBAIKAN
1	9 September 2016	C. Deo P. Bramantio Y. Kevin S.P.	Versi pertama dokumen B100
2	1 Mei 2017	C. Deo P. Bramantio Y. Kevin S.P.	Revisi pertama B100

Proposal Proyek Pengembangan Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*

1 Pengantar

1.1 RINGKASAN ISI DOKUMEN

Dokumen B100 ini merupakan proposal mengenai proyek pengembangan Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System* yang implementasinya akan diaplikasikan di Indonesia. Dokumen B100 ini terbagi dalam tiga bab. Bab pertama berisi pengantar yaitu ringkasan isi dokumen, tujuan penulisan/kegunaan dokumen, referensi, dan daftar singkatan. Bab kedua berisi tentang masalah dan tujuan, analisis umum, karakteristik produk, cost estimate, analisa finansial, skenario pemanfaatan produk, skenario dan perencanaan pengembangan proyek, dan kesimpulan dan ringkasan. Terakhir bab ketiga terdapat lampiran yang berisi data anggota tim, rencana kerja, dan daftar perangkat keras yang diperlukan selama proses pengerjaan proyek tugas akhir.

1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen

Tujuan penulisan dokumen B100 ini adalah sebagai berikut:

1. sebagai gambaran umum dari pengerjaan proyek selaku tugas akhir Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*;
2. sebagai penyampai gagasan dan konsep dari proyek selaku tugas akhir Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*;
3. sebagai dokumentasi dalam pengerjaan proyek selaku tugas akhir Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*.

Dokumen B100 ini ditujukan kepada dosen pembimbing TA1617-01-069 dan tim tugas akhir Program Studi Teknik Elektro ITB 2016 sebagai bahan penilaian tugas akhir.

1.3 REFERENSI

- [1] Rudloff, Alexander, *German-Indonesian Tsunami Early Warning System (GITEWS) Decision Support System (DSS)*, Deutschen Zentrums für Luft- und Raumfahrt (DLR): Köln (2010)
- [2] Sieh, K., & Natawidjaja, D. (2000). *Neotectonics of the Sumatran fault*, Indonesia. *Journal of Geophysical Research*, 105, 28295–28326
- [3] Lay, Thorne et al. *The Great Sumatra-Andaman Earthquake of 26 December 2004*, *Science* 20 May 2005: Vol. 308, Issue 5725, pp. 1127-1133

1.4 DAFTAR SINGKATAN

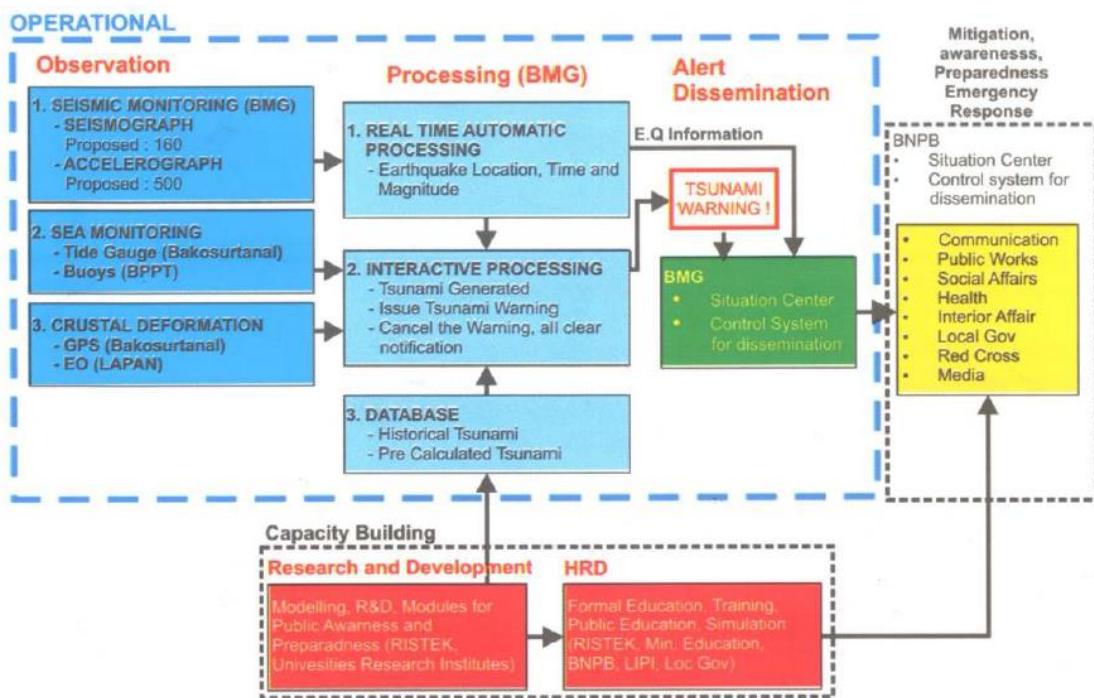
SINGKATAN	ARTI
APBN	Anggaran Pendapatan dan Belanja Negara
BMKG	Badan Meteorologi, Klimatologi, dan Geofisika
BPPT	Badan Pengkajian dan Penerapan Teknologi
DSS	Decision Support System
GPS	Global Positioning System
GITEWS	German Indonesian Tsunami Early Warning System
InaTEWS	Indonesia Tsunami Early Warning System
ITB	Institut Teknologi Bandung
LAPAN	Lembaga Penerbangan dan Antariksa Nasional
LIPI	Lembaga Ilmu Pengetahuan Indonesia
Pemda	Pemerintah daerah
PPMB	Pusat Penelitian Mitigasi Bencana
PPTIK	Pusat Penelitian Teknologi, Informasi dan Komunikasi

2 DEVELOPMENT PROJECT PROPOSAL

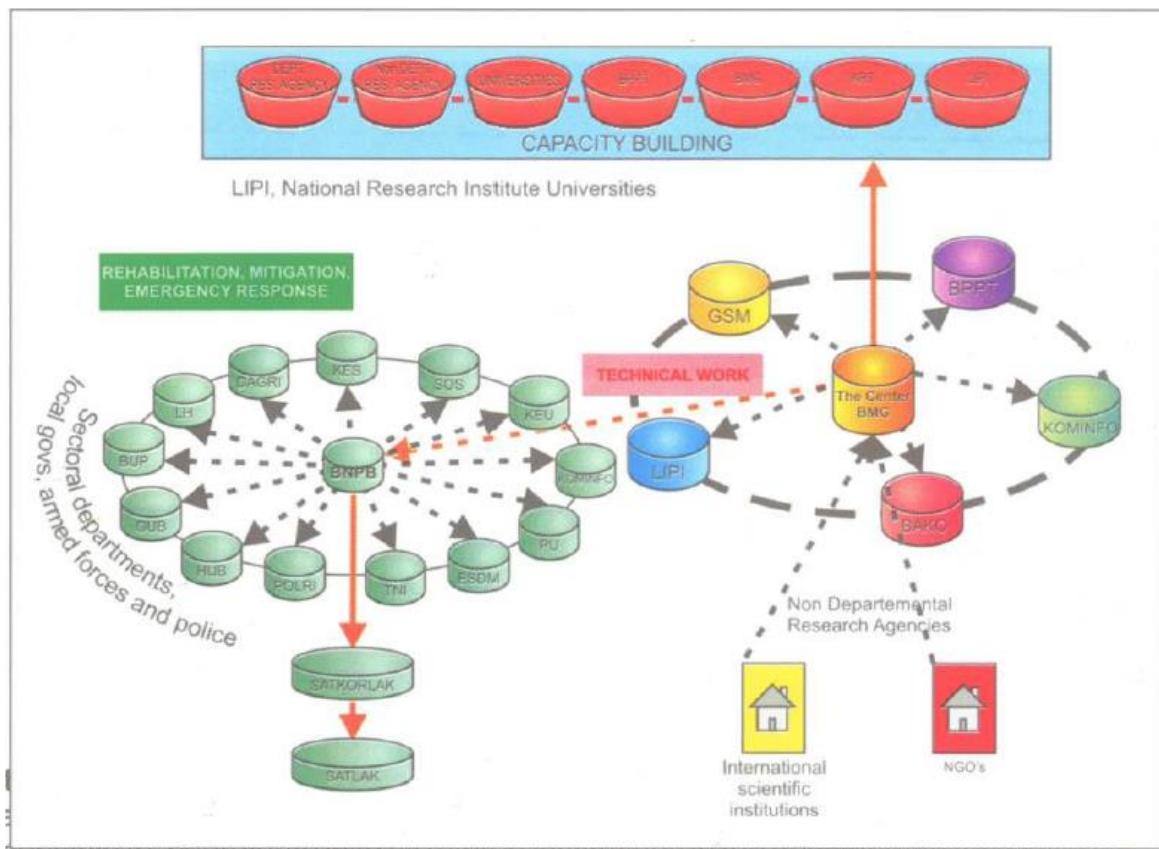
2.1 MASALAH DAN TUJUAN

Dapat dikatakan bahwa Indonesia merupakan negara yang memiliki potensi gempa bumi dan tsunami yang cukup besar. Hal ini disebabkan karena Indonesia berada di zona yang memiliki aktivitas seismik yang tinggi. Tepatnya, pesisir barat Sumatra dan pesisir selatan Jawa terletak berdekatan dengan zona subduksi yang merupakan bagian dari *Sunda Arc*. Zona subduksi di dekat Sumatra juga memiliki nama sendiri, yakni *Great Sumatran Fault*. Sejak tahun 1991 sampai 2009 telah dicatat sebanyak 30 gempa bumi dan 14 tsunami yang bersifat merusak, dengan jumlah korban total lebih dari 300 ribu jiwa. Sehingga dapat disimpulkan bahwa tsunami menghantam pesisir Indonesia rata-rata sebanyak sekali setiap dua tahun, dengan rata-rata setiap tahunnya ada 30 ribu orang yang kehilangan nyawa akibat tsunami. Untuk menanggapi kondisi Indonesia tersebut, pemerintah Indonesia sudah membuat InaTEWS sejak tahun 2008.

InaTEWS (*Indonesia Tsunami Early Warning System*) dibangun dengan kerjasama dengan pemerintah republik federal Jerman, melalui Deutschen Zentrums für Luft- und Raumfahrt (DLR), atau pusat urusan langit Jerman. InaTEWS didasarkan pada GITEWS (*German Indonesian Tsunami Early Warning System*). Pada awalnya, GITEWS dioperasikan secara bersama oleh pihak Jerman dan Indonesia. Namun, pada tahun 2008, operasi dari sistem peringatan dini tsunami diserahkan sepenuhnya pada pihak Indonesia, dan namanya berubah menjadi nama yang digunakan sekarang yaitu InaTEWS.



Gambar 1 Diagram GITEWS



Gambar 2 Skenario Disemenasi Bencana

InaTEWS memiliki 3 komponen, yaitu Komponen Operasional, Komponen Mitigasi dan Respon Keadaan Darurat, serta Komponen Pembangunan Kapasitas.

1. Komponen Operasional

Komponen Operasional mengatasi aktivitas monitoring, pemrosesan, analisis, persiapan peringatan, dan diseminasi.

2. Komponen Mitigasi dan Respon Keadaan Darurat

Komponen Mitigasi dan Respon Keadaan Darurat mengatasi respon darurat terhadap bencana, pendidikan publik, meningkatkan kewaspadaan publik terhadap bencana, persiapan logistik dan tempat perlindungan, peta evakuasi, dan pelatihan lapangan

3. Komponen Pembangunan Kapasitas

Komponen Pembangunan Kapasitas memberikan bantuan melalui riset dan meningkatkan kapasitas dari sumber daya manusia.

Sedangkan pada komponen operasional, pembagian kerja dibagi menjadi tiga, yaitu Observasi, Pemrosesan Data, dan Diseminasi.

1. Observasi

Pada bagian ini dilakukan monitoring aktivitas seismik yang dilakukan oleh BMG, monitoring keadaan laut yang dilakukan oleh Bakosurtanal dan BPPT, serta monitoring perubahan bentuk crust bumi yang dilakukan oleh Bakosurtanal dan LAPAN. Data-data yang diperoleh dari pembacaan sensor dikirimkan secara real-time ke BMG agar dapat dilakukan pemrosesan data.

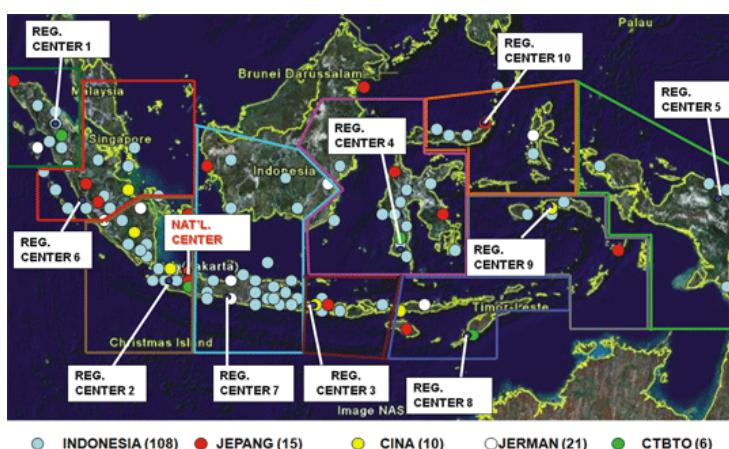
2. Pemrosesan Data

Setelah dilakukan penerimaan data dari proses observasi, dilakukan pemrosesan realtime secara automatis. Dalam pengambilan keputusan peringatan tsunami, digunakan juga database yang berisi mengenai data yang diperoleh saat bencana tsunami yang sudah terjadi.

3. Diseminasi

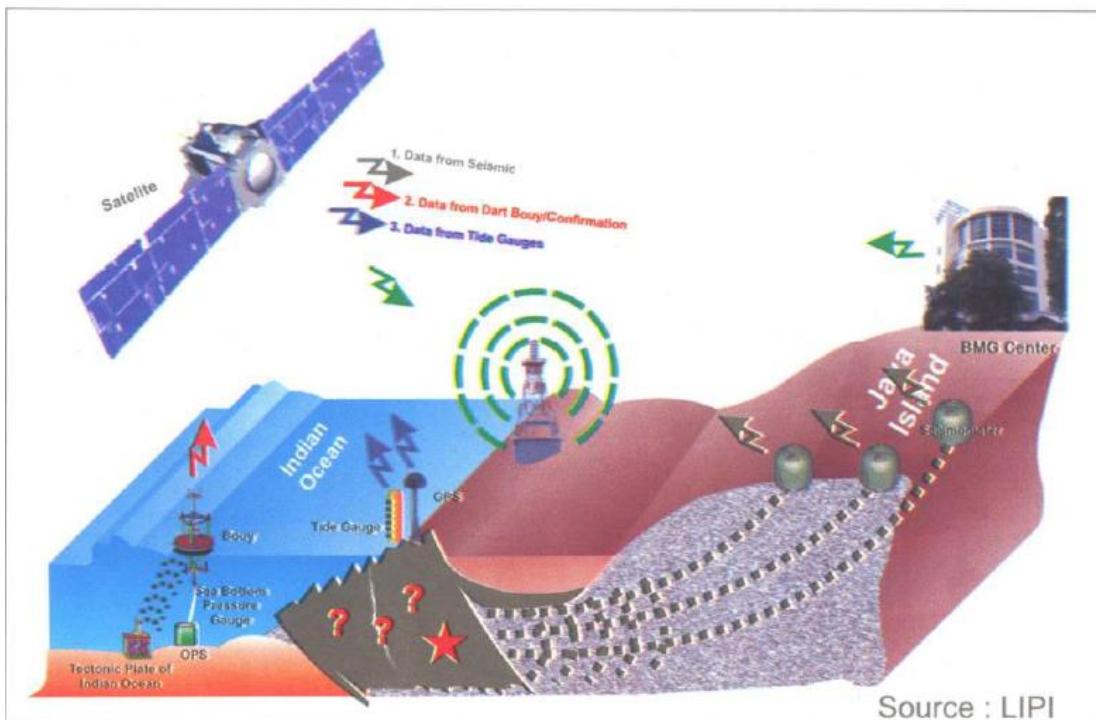
Ketika hasil dari proses pemrosesan data mengindikasikan terjadinya tsunami, peringatan tsunami segera diinformasikan kepada lembaga yang termasuk kedalam komponen mitigasi dan respon keadaan darurat.

Agar proses diseminasi berjalan efektif dan efisien diperlukan adanya sistem yang dapat mendapatkan data yang akurat secara real time agar dapat segera dianalisis secara real-time juga. Keputusan terkait peringatan tsunami harus dapat diinformasikan secara cepat ke lembaga yang berwenang untuk melakukan diseminasi setelah terjadinya gempa. Selain itu dibutuhkan juga sistem yang menggunakan sensor lebih murah dari seismometer karena sistem ini akan diimplementasikan di seluruh wilayah Indonesia yang lokasinya berdekatan dengan lempeng tektonik sehingga membutuhkan sensor dengan jumlah cukup banyak.



Gambar 3 Peta Jaringan Seismik Indonesia

Jaringan Seismik Indonesia terdiri dari 160 broadband seismometer, 500 accelerometer dan akan dikelompokan kedalam 10 Regional Center. Dengan jumlah sensor tersebut dan jarak tiap sensor ± 100 km, maka dalam 3 menit pertama sumber gempabumi yang terjadi di wilayah Indonesia dapat ditentukan lokasinya. Namun, sensor tersebut adalah seismometer yang mahal, dan tidak semua seismometer yang ada di jaringan seismik Indonesia dimiliki oleh Indonesia, seperti yang bisa dilihat di gambar diatas. Selain itu, hanya ada sekitar 20 sensor yang diletakkan pada pesisir barat Sumatera dan pesisir selatan Jawa, yang cukup sedikit untuk garis pantai sepanjang itu.



Gambar 4 Cara Kerja Sensor Seismik dan Buoy

Dengan adanya masalah-masalah yang telah disebutkan, diperlukannya Sistem Jaringan Deteksi Gempa dan Tsunami Decision Support System yang bertujuan untuk membuat keputusan terkait peringatan tsunami dan gempa menjadi lebih efisien, akurat, cepat, dan menghabiskan dana yang lebih sedikit jika dibandingkan dengan menggunakan seismometer.

2.2 ANALISIS UMUM

Dengan adanya Sistem Jaringan Deteksi Gempa dan Tsunami Decision Support System, proses diseminasi akan berjalan menjadi lebih efektif. Dengan diseminasi yang baik, masyarakat akan menjadi lebih waspada dan siap terhadap bencana gempa dan tsunami. Proses diseminasi yang baik juga akan membuat masyarakat merasa lebih aman karena adanya sistem peringatan gempa dan tsunami yang baik.

Selain dapat membuat proses diseminasi menjadi lebih efektif, dengan adanya Sistem Jaringan Deteksi Gempa dan Tsunami Decision Support System, data yang diperoleh dari sistem ini juga dapat dimanfaatkan untuk beberapa hal lain selain untuk sebagai bahan dasar pengambilan keputusan terkait peringatan gempa dan tsunami. Dengan adanya data

tersebut, dapat diperoleh data lokasi rawan gempa yang dapat digunakan sebagai dasar kebijakan yang diambil pemerintah dan sebagai dasar strategi bisnis ataupun kebijakan dari suatu perusahaan.

2.3 PRODUCT CHARACTERISTIC

2.3.1 Fitur Utama

Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System ini terdiri dari sistem multi komponen terintegrasi dengan jaringan nirkabel yang menggunakan data pengukuran secara aktual dari berbagai sensor yang digunakan dan akan dibandingkan dengan model gempa dan tsunami yang hasilnya disajikan dalam bentuk estimasi lokasi pusat gempa dan bahaya tsunami yang akan datang. Hasil ini akan diolah pada Decision Support System (DSS) yang nantinya akan memberikan bantuan ketika dan sesudah terjadi gempa dan tsunami.

2.3.2 Fitur Dasar

2.3.2.1 Sistem Sensor

2.3.2.1.1 Sistem Pengamatan Gempa

Menggunakan sensor gerakan yang dikombinasikan dengan jaringan nirkabel. Sensor ini akan disebar lalu akan memberikan hasil pengukuran aktual menuju sistem pengamatan gempa yang akan menghitung parameter dari sumber gempa. Sistem ini akan memberikan hasil dalam hitungan menit.

2.3.2.1.2 Instrumentasi Pengukuran di Perairan

Karena tidak setiap peristiwa gempa menimbulkan tsunami, maka diperlukan data lain yang menentukan apakah akan terjadi tsunami atau tidak. Oleh karena itu dibutuhkan sensor gerakan pada perairan dengan fungsi yang berbeda dengan yang digunakan dalam Sistem Pengamatan Gempa sehingga berfungsi mendeteksi adanya tsunami. Sensor ini diintegrasikan dengan GPS dan datanya akan diintegrasikan dengan Sistem Pengamatan Gempa.

2.3.2.2 Sistem Permodelan Tsunami

2.3.2.2.1 Permodelan Sumber Gempa

Kekuatan gempa dapat bervariasi tergantung dari energi dan letaknya. Apabila gempa terletak di daerah pantai maka diperlukan permodelan bagaimana gempa akan merambat dari pusat gempa menuju daratan.

2.3.2.2.2 Permodelan Tsunami

Diperlukan permodelan yang detail tentang propagasi tsunami sehingga bisa diprediksi ketinggian air dan luasan yang akan diterjang tsunami.

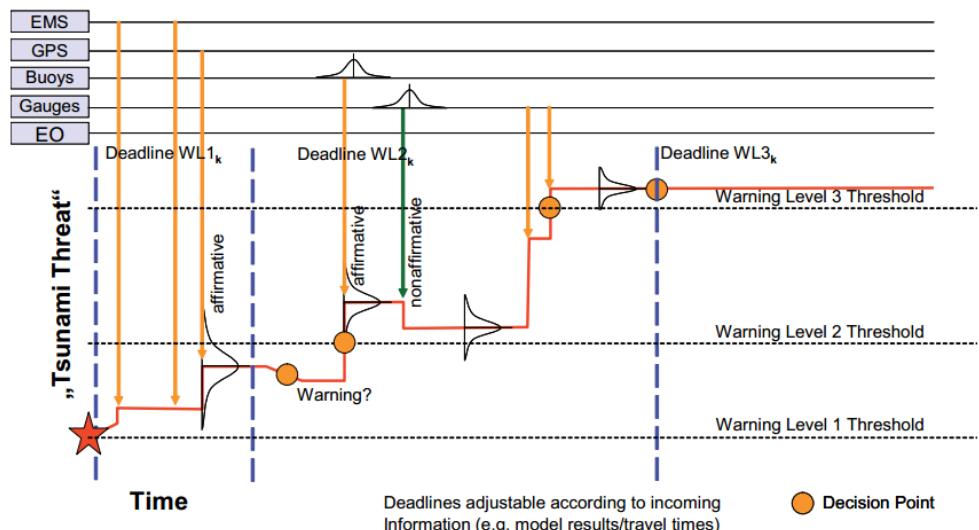
2.3.2.2.3 Permodelan Terjangan Tsunami di Darat.

Air yang memasuki daerah daratan akibat dari Tsunami akan dimodelkan sehingga bisa menentukan perencanaan perlindungan awal, termasuk dengan evakuasi populasi masyarakat sekitar.

Semua permodelan yang ada akan sulit untuk disimulasikan dalam jaringan sehingga permodelan dilakukan secara luar jaringan dengan menggunakan skenario yang mungkin terjadi.

2.3.2.3 Decision Support System

Dalam Decision Support System dibutuhkan data yang aktual dan terpercaya. Hal itu dapat ditunjukkan dalam diagram berikut:



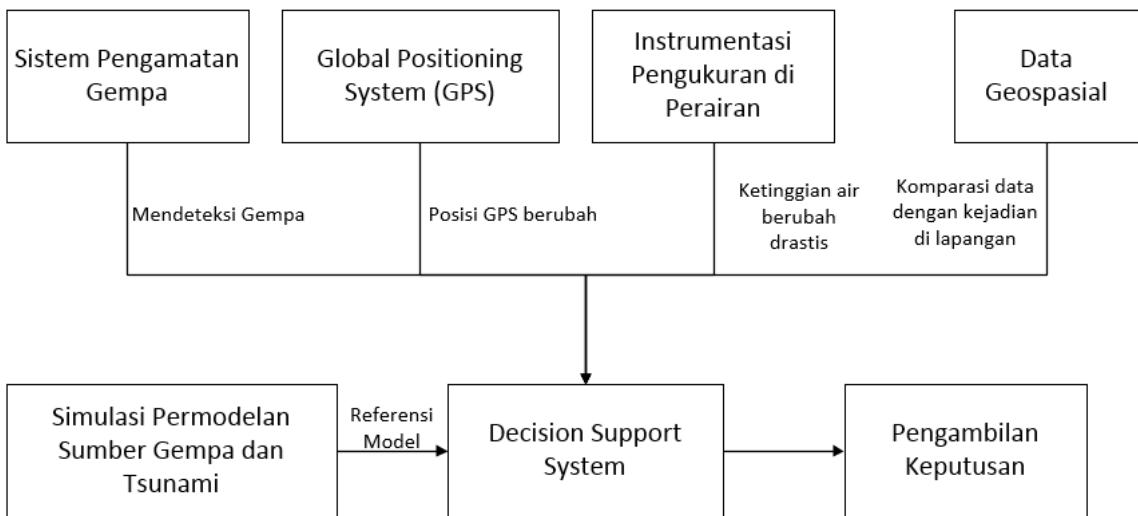
Gambar 5 Diagram alur timing dari contoh Decision Support

Lingkaran (Decision Point) pada diagram alur diatas merupakan titik dimana sistem membantu memberikan pilihan keputusan kepada pihak berwenang.

2.3.3 Fitur Tambahan

2.3.3.1 Penyimpanan Data Geospasial dalam Internet

Informasi yang ada pada saat terjadinya gempa dan tsunami maupun sesudahnya dilengkapi dengan data wilayah yang bersangkutan. Maksudnya adalah semua data yang diperoleh merujuk kepada wilayah tertentu. Data ini akan di akuisisi di dalam Internet sehingga bisa ditampilkan dalam bentuk visualisasi secara dalam maupun luar jaringan. Setiap masuknya data yang baru akan memperbaharui atau melengkapi data yang lama.



Gambar diagram flow dari Decision Support System

2.3.4 Sifat solusi yang diharapkan

- **Lebih murah** karena menggunakan sensor digital yang umum digunakan oleh orang awam;
- **Jangkauan pembacaan data yang lebih jauh** karena harga yang murah sehingga dapat membeli sensor yang lebih banyak;
- **Aktual** karena terintegrasi dengan sistem secara dalam jaringan;
- **Informatif** karena sudah dilakukan permodelan pra-bencana.
- **Berkesinambungan** karena akan digunakan oleh operator dari pihak yang berwenang.

2.4 COST ESTIMATE

Belanja

Sebelum bisa membuat sebuah perkiraan dana yang dibutuhkan untuk proyek ini, terlebih dahulu kami perlu menguraikan desain awal dari sistem yang akan dibuat. Rincian sistem tersebut adalah sebagai berikut:

- Sistem pemanatau

Sistem pemantau ini terbagi menjadi bagian darat dan laut. Pada bagian seismik, ada 160 seismometer pita lebar, 500 sensor akselerasi yang akan dikelompokkan menjadi 10 pusat regional. Bagian Bakosurtanal terdiri dari *GPS* yang berjumlah 40 buah dan 80 sistem pemantau gelombang pasang. Di laut, akan dipasang sekitar 10 buah *buoy* yang menjadi detektor tsunami. Detil *buoy* ada pada gambar dibawah



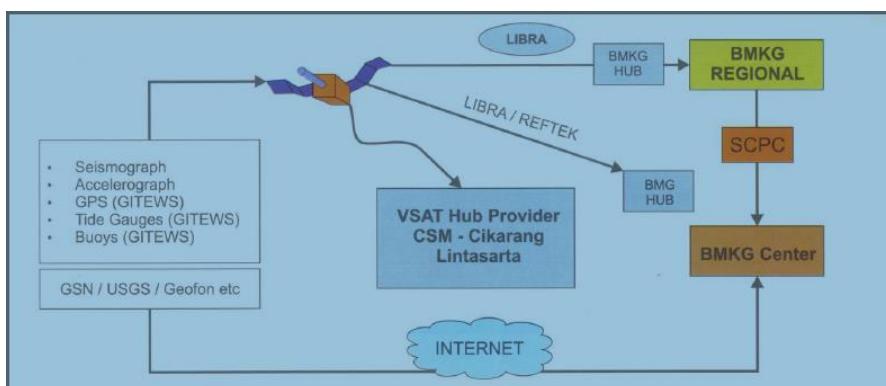
Gambar 6 Peta Persebaran Buoy di Laut Indonesia

- **Sistem Pemrosesan**

Sistem ini terdiri dari 10 pusat regional dan satu pusat nasional untuk pemantauan seismik, 1 pusat pemantauan gelombang pasang, 1 pusat *buoy*, dan 1 pusat pemantauan GPS.

- **Telekomunikasi**

Komunikasi pada sistem ini ada dua arah, yaitu pengumpulan informasi dan diseminasi. Sistem komunikasi tersebut menggunakan teknologi VSAT dengan sistem LIBRA, Reftec, dan CSM yang disewa oleh BMKG. Skema komunikasi tersebut adalah sebagai berikut:

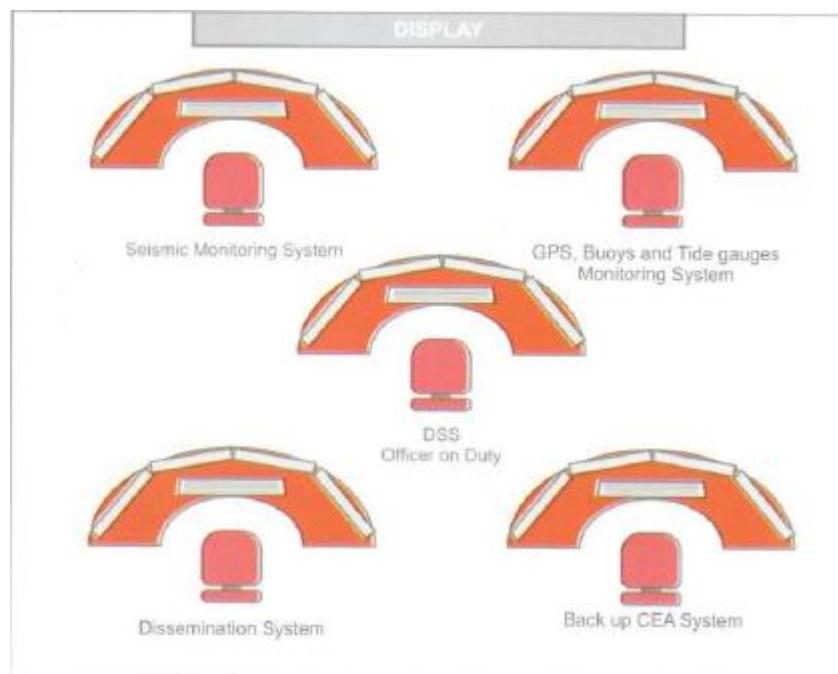


Gambar 7 Diagram Telekomunikasi Sistem Seismik

- **Pusat Peringatan Tsunami Nasional**

Gedung ini akan dibawahi oleh BMKG dan berisi fasilitas pemrosesan data seismik dari berbagai negara dan akan berfungsi sebagai pusat dari *Decision Support System*. Pusat ini akan dikelola oleh staff yang akan memberi informasi yang kritis seputar tsunami saat terjadi gempa, sehingga membantu dalam

pengambilan keputusan. Contoh peran staff pada Pusat Peringatan ini adalah sebagai berikut:



Gambar 8 Pembagian Tugas di Ruangan Kontrol Sistem Seismik

Dari rancangan sistem tersebut, estimasi biayanya adalah sebagai berikut:

- Biaya Awal

Barang	Kuantitas	Harga Satuan	Jumlah
Seismometer	160	Rp450,000,000	Rp72,000,000,000
DART Tsunameter Buoy	12	Rp5,250,000,000	Rp63,000,000,000
Tide Gauge Sensor	80	Rp54,000,000	Rp4,320,000,000
Sensor GPS	40	Rp19,500,000	Rp780,000,000
Stasiun Pemantauan	16	Rp50,000,000,000	Rp800,000,000,000
Pengembangan (Non-Recurring Engineering)			Rp100,000,000,000
Infrastruktur			Rp100,000,000,000
Total			Rp1,140,100,000,000

- Biaya per Tahun

Keperluan	Jumlah
Perawatan Sensor	Rp200,000,000
Penggunaan Satelit Komunikasi	Rp300,000,000
Upah Pegawai	Rp250,000,000
Perawatan Infrastruktur	Rp300,000,000
Total	Rp1,050,000,000

Anggaran

Untuk mendapatkan estimasi anggaran yang cocok dengan kebutuhan, kami perlu mengasumsikan beberapa hal sebagai berikut:

1. Pertumbuhan ekonomi Indonesia : 5,3%
2. Inflasi : 4,7%
3. Tingkat Bunga SPN : 5,5%
4. Nilai tukar dolar : Rp.13.900,- (sedang mengalami kenaikan)

Opsi pertama sumber dana untuk proyek ini adalah APBN negara. Pada tahun 2016, dari total belanja negara sebesar 2.095 triliun rupiah, sebesar 4,5 triliun rupiah digunakan untuk bantuan sosial dan penganggulangan bencana. Anggaran tersebut termasuk ke dalam biaya non-kementerian/lembaga. Dengan menggunakan anggaran negara, jatah untuk bencana lain menjadi berkurang, namun proyek ini adalah proyek yang *sustainable* sehingga hanya perlu pengeluaran besar saat pertama, dengan biaya perawatan yang cukup kecil. Selain itu, menggunakan APBN menghindarkan dari bunga yang bila digabung dengan inflasi, bisa menyebabkan kenaikan biaya yang cukup signifikan.

Opsi kedua adalah meminjam ke *International Monetary Fund*(IMF). Opsi ini sebaiknya dihindari karena tingkat bunga *Special Drawing Rate* yang cukup besar terutama untuk mata uang dollar Amerika, yaitu sekitar 15%. Nilai ini jauh lebih dari tingkat bunga SPN di dalam negeri. Bunga SDR tersebut juga jauh melebihi tingkat inflasi sehingga negara akan perlu mengeluarkan biaya yang besar hanya untuk bunga. Hal ini juga didukung dengan fakta bahwa IMF menolak untuk meminjamkan uang ke suatu negara untuk keperluan proyek tertentu, sehingga Indonesia perlu meminjam dengan jumlah yang lebih besar dari seharusnya dan menggunakan sisanya untuk keperluan lain yang didukung IMF seperti kebijakan yang mendukung stabilitas ekonomi.

Opsi terakhir adalah meminjam ke *Development Bank* seperti ADB yang telah mendukung program Indonesia seperti pendidikan vokasi pada tahun 2008. Bunga ADB lebih kecil dari IMF, yaitu sekitar 8%. Namun, ADB cukup menuai kritik karena besarnya pengaruh Jepang dan Amerika Serikat, sehingga negara kita mungkin mengorbankan sedikit kebebasan demi proyek ini.

2.5 ANALISIS FINANSIAL

Proyek ini bukanlah suatu produk yang didesain untuk mendapatkan keuntungan, namun suatu pencegahan kerugian. Tidak ada proyeksi keuntungan oleh pemerintah Indonesia dari proyek ini, jadi kami akan melakukan analisis dari sisi kerugian yang bisa dihindari dengan sistem ini, dengan mengambil data dari bencana yang sudah terjadi. Beberapa perkiraan dan asumsi yang kami ambil adalah sebagai berikut:

1. GDP per kapita Indonesia sebesar 3.475 USD
2. Nilai tukar dolar : Rp.13.900,- (sedang mengalami kenaikan)
3. Tingkat Bunga SPN : 5,5%

Pertama akan dibahas seberapa besar pengaruh tsunami terhadap Indonesia. Indonesia terletak di perbatasan antara lempeng Eurasia dan lempeng Indo-Australia dimana ada zona subduksi di lepas pantai Sumatera dan Jawa. Zona ini merupakan daerah rawan gempa dan bila lokasi episentrum tidak dalam ($<70\text{km}$) digabung dengan intensitas gempa yang besar (>7.0 SR) maka gempa tersebut sangat berpotensi untuk menghasilkan tsunami. Dari tahun 1991 hingga 2009 terdapat 30 gempa besar dan 14 diantaranya menghasilkan tsunami yang cukup merusak. Tsunami Aceh tahun 2004 adalah hasil dari gempa terbesar ketiga yang pernah tercatat, dan kemungkinan terjadinya adalah sekitar satu kali dalam seratus tahun. Namun, tsunami yang lebih kecil dari itu cukup sering terjadi, seperti di Jawa Timur (1994), Flores (1991), Biak(1996), Nias (2005) dan Bengkulu (2007). Bila diambil rata-rata, maka tsunami yang merusak terjadi sekali tiap dua tahun.

Berikutnya kami akan membahas kerugian yang ditimbulkan akibat tsunami. Kejadian di Aceh pada tahun 2004 memakan korban jiwa sebanyak 230.000 sampai 280.000, dimana sekitar 160.000 diantaranya adalah orang Indonesia. Jumlah orang yang hilang sebanyak 37.000, dengan 500.000 lainnya kehilangan tempat tinggal. Bila diasumsikan bahwa 100.000 saja dari jumlah korban meninggal adalah penduduk usia produktif, maka kerugian negara dari kehilangan jiwa bisa ditaksir dengan mengalikannya dengan GDP per kapita, yang menghasilkan angka 5,2 triliun rupiah per tahun. Jika dibandingkan dengan biaya pembuatan sistem yang berkisar 1,2 triliun, secara finansial negara masih akan untung bahkan jika sistem hanya efektif 50%.

Dalam Background Paper on the Benefits and Costs of Early Warning Systems for Major Natural Hazards, oleh Teisberg dan Weiher, tsunami ditempatkan di bagian tengah pada analisis tentang keuntungan membuat sistem peringatan dini untuk bencana tertentu, dengan rincian sebagai berikut:

Scores by Life Saving Benefits	Frequency	Predictability	Severity	Potential	
			(loss of life)	Life Saving	Summary
Cyclones	5	5	4	5	4.7
Floods	5	5	2	3	3.5
Volcanoes	2	5	3	5	3.5
Heat	3	5	2	3	3.1
Tsunamis	1	3	5	4	2.8
Tornadoes	2	3	2	3	2.4
Drought	2	2	1	1	1.4
Earthquakes	3	0	5	0	0.0

Namun, bila dilihat data di Indonesia, frekuensi tsunami cukup banyak, meski yang separah kejadian di Aceh mungkin hanya sedikit. Karena itu, di Indonesia cukup menguntungkan bila dibuat sistem peringatan dini terhadap tsunami, karena skor lain cukup baik. Pendekatan lain untuk cost-benefit analysis ini menggunakan teorema present value. Akan diambil data statistik periode 1990-2010 untuk mencari rata-rata korban meninggal akibat tsunami di indonesia, dan dengan memasang suatu harga pada korban meninggal dapat dianalisis kebermanfaatan dari sistem ini.

Tahun	Lokasi	Korban
2004	Aceh	227,898
2006	Bantul-Yogyakarta	5,749
1992	Flores	2,500
2005	Sumatera Utara	1,313
2009	Sumatera Selatan	1,117
2005	Nias	1,314
2006	Jawa Barat	668
2007	Bengkulu	23

Dari data tersebut didapatkan bahwa total ada sekitar 240 ribu korban jiwa akibat tsunami di Indonesia selama periode 1990-2010 yaitu selama 20 tahun. Bila dirata-rata ada 12 ribu orang yang meninggal setiap tahunnya akibat tsunami. Untuk melakukan analisis finansial, kami menggunakan angka \$129.000 untuk setiap korban jiwa, yang merupakan hasil penelitian Stefanos Zeinos dari Stanford University. Dengan begitu sistem ini mempunyai keuntungan sebesar 1,5 miliar dollar setiap tahunnya, yang bila dikonversi ke rupiah menjadi 23 trilyun rupiah. Dengan teorema present value bisa dihitung nilai sekarang dari keuntungan tersebut, dengan asumsi sistem bertahan selama 20 tahun dengan bunga konstan

$$PV = \frac{C}{(1 + i)^n}$$

Dari hasil perhitungan tersebut didapat nilai present value sebesar 157 triliyun rupiah, yang mendekati 150 kali biaya awal, dengan mengabaikan biaya tahunan yang kurang signifikan. Namun perlu diketahui bahwa data korban meninggal kebanyakan berasal dari tsunami Aceh yang terjadi mungkin hanya 100 tahun sekali. Bila kita mengabaikan tsunami 2004 yang dianggap *rogue value*, maka didapat 634 orang yang meninggal setiap tahun karena tsunami. Dengan pendekatan yang sama, didapatkan manfaat sebesar 1,1 triliyun rupiah per tahun. Present value dari manfaat tersebut selama 20 tahun sebesar 7,7 triliyun rupiah, hampir 7 kali biaya awal sistem. Juga perlu dicatat bahwa \$129.000 untuk nyawa manusia cukup rendah, karena sebagai contoh *United States Environmental Protection Agency* memasang harga 6 juta dolar pada setiap nyawa manusia yang hilang dalam analisis mengenai polusi udara. Sehingga dapat disimpulkan bahwa membangun sistem InaTEWS merupakan investasi yang sangat menguntungkan.

2.6 SKENARIO PEMANFAATAN PRODUK

Perihal	Pihak
Pembeli	<ul style="list-style-type: none">- Badan Meteorologi, Klimatogi dan Geofisika (BMKG)- Pemerintah Daerah (Pemda)
Pengembangan	<ul style="list-style-type: none">- Pusat Penelitian Teknologi, Informasi dan Komunikasi Institut Teknologi Bandung (PPTIK ITB)- Pusat Penelitian Mitigasi Bencana Institut Teknologi Bandung (PPMB ITB)- Analis Big Data
Profit	<ul style="list-style-type: none">- Masyarakat- Pemerintah

Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System akan dipakai oleh BMKG (Badan Meteorologi, Klimatogi dan Geofisika) sehingga pembelian, dan pengoperasian dari produk ini adalah dari BMKG. Selain itu, terdapat kerja sama antara BMKG dengan Pusat Penelitian Teknologi, Informasi dan Komunikasi Institut Teknologi Bandung (PPTIK ITB) yang berfungsi sebagai peneliti maupun fasilitator bagian data, Pusat Penelitian Mitigasi Bencana (PPMB) ITB yang berfungsi sebagai peneliti, dan juga Badan Pengkajian dan Penerapan Teknologi sebagai pengkaji produk ini. Selain itu, produk ini memiliki nilai jual berupa data yang bisa digunakan oleh analis yang membutuhkan data sebagai bahan penelitian. Apabila produk ini diimplementasikan, maka yang diuntungkan adalah masyarakat daerah rawan gempa dan tsunami, dan juga pemerintah daerah.

2.7 SKENARIO PENGEMBANGAN PRODUKSI DAN PEMASARAN

Perihal	Pihak
Produksi	<ul style="list-style-type: none">- Vendor Sensor- Layanan penyedia server
Investasi	<ul style="list-style-type: none">- Anggaran Pendapatan dan Belanja Negara (APBN)- Sumbangan dari pihak luar negeri
Pemasaran	<ul style="list-style-type: none">- Badan Meteorologi, Klimatogi dan Geofisika (BMKG)
Model Bisnis	<ul style="list-style-type: none">- Tender Pemerintah Pusat
Perijinan	<ul style="list-style-type: none">- Badan Pengkajian dan Penerapan Teknologi (BPPT)

Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System akan diproduksi oleh vendor penyedia sensor sebagai pembuat hardware dan juga layanan penyedia server sebagai pusat penyimpanan data dalam jaringan. Produk ini diinvestasikan oleh Pemerintah Pusat dengan menggunakan Anggaran Pendapatan dan Belanja Negara (APBN) karena memang sudah ada Undang-Undang yang berlaku dan juga investasi dari pihak asing untuk meneliti gempa menggunakan teknologi ini. Pemasaran produk ini dilakukan oleh BMKG karena sebagai pihak yang menggunakan, maka juga dibutuhkan usaha untuk mencari dana tambahan sebagai biaya pengembangan teknologi ini. Model bisnis yang digunakan adalah dengan menawarkan spesifikasi yang dibutuhkan melalui Pemerintah Pusat sehingga ada pihak swasta yang mengambil proyek ini dan kemudian ikut mengembangkan produk dengan menjadi supplier. Perijinan dari produk ini dilakukan oleh BPPT sehingga bisa dikontrol perkembangan dan juga keamanan dari teknologi ini.

2.8 KESIMPULAN DAN RINGKASAN

Sistem yang akan dibuat bertujuan untuk meningkatkan keefektivitasan diseminasi gempa dan tsunami di Indonesia dengan menggunakan biaya yang semurah mungkin untuk memudahkan melakukan decision support kepada pihak yang berwenang apabila terjadi gempa dan tsunami, sehingga meminimalisasi jumlah korban bencana. Pemerintah juga telah mengalokasikan biaya untuk mengatasi korban bencana dan sistem ini bisa menjadi solusi untuk masalah menghadapi gempa dan tsunami di Indonesia. Dengan adanya sistem ini, masyarakat akan lebih siap dan waspada akan adanya gempa dan tsunami.

Produk yang digunakan dalam sistem ini berupa penginderaan atau sensor yang bisa mendeteksi gempa dan tsunami secara langsung dan sistem yang mengintegrasikan semua penginderaan yang ada. Hasilnya akan diolah dan dijadikan bahan keputusan oleh pihak yang berwenang untuk mengambil keputusan dengan cepat. Sistem ini dilengkapi dengan jaringan sehingga cakupan sistem yang bisa mendeteksi gempa dan tsunami akan lebih luas, tersebar merata, dan memberikan respon yang cepat.

Saat ini, korban bencana akibat gempa dan tsunami dalam 20 tahun terakhir mencapai lebih dari 225.000 orang. Pemerintah bertanggung jawab dalam mengamankan penduduk yang ada di Indonesia dari bencana yang ada, sehingga dengan adanya sistem ini, angka korban yang mungkin terjadi dalam 20 tahun selanjutnya akan menuju angka minimal.

3 Lampiran

3.1 Data Anggota Tim

Anggota 1

Identitas	
Nama	Christoporus Deo Putratama
Nomor Induk Mahasiswa	13213008
Program Studi	Teknik Elektro ITB
Latar Belakang	
Tempat lahir	Bandung
Tanggal lahir	18-08-1995
Alamat (libur)	Jl. Muntilan no.41, Komp. Pharmindo, Kel. Melong, Kec. Cimahi Selatan, Cimahi 40534
Pendidikan	SD Santa Angela Bandung (2001-2007) SMP Santa Angela Bandung (2007-2010) SMA Negeri 3 Bandung (2010-2013) Institut Teknologi Bandung (2013-sekarang)
Estimasi kapasitas kerja	
Alamat (masa kuliah)	Jl. Muntilan no.41, Komp. Pharmindo, Kel. Melong, Kec. Cimahi Selatan, Cimahi 40534
Status tempat tinggal	Rumah
Mobilitas (skala 5)	4
Keahlian	Programming : C, C#, Matlab, VHDL, Arduino, CVAVR, Software : Microsoft Office, Altium Designer, EagleCAD, LT Spice, CorelDraw, Visual Studio, Autocad Inventor, Quartus
Jumlah kredit yang diambil untuk Semester I = 2016/2017 di luar EL4096	
Perkiraan alokasi waktu untuk seluruh kegiatan non-akademis (per minggu)	
Perkiraan alokasi waktu untuk seluruh kegiatan non-akademis (per minggu)	

Anggota 2

Identitas	
Nama	Bramantio Yuwono
Nomor Induk Mahasiswa	13213126
Program Studi	Teknik Elektro ITB
Latar Belakang	
Tempat lahir	Jakarta
Tanggal lahir	26-07-1995
Alamat (libur)	Jalan Betung Raya no 16 RT 11 RW 05 Pondok Bambu Jakarta Timur
Pendidikan	SD 1 Pondok Jakarta Timur (2001-2003) SD Putra 1 Jakarta Timur (2001-2003) SMP Negeri 109 Jakarta Timur (2007-2010) SMA Negeri 81 Jakarta Timur (2010-2013) Institut Teknologi Bandung (2013-sekarang)
Estimasi kapasitas kerja	
Alamat (masa kuliah)	Jalan Plesiran no. 19, Bandung
Status tempat tinggal	Kost
Mobilitas (skala 5)	4
Keahlian	Programming : C, C#, Matlab, VHDL, Software : Microsoft Office, Altium Designer, LT Spice, CorelDraw, Visual Studio, Autodesk Inventor, Quartus, Proteus Professional
Jumlah kredit yang diambil untuk Semester I = 2016/2017 di luar EL4096	
Perkiraan alokasi waktu untuk seluruh kegiatan non-akademis (per minggu)	
Perkiraan alokasi waktu untuk seluruh kegiatan non-akademis (per minggu)	

Anggota 3

Identitas	
Nama	Kevin Shidqi Prakoso
Nomor Induk Mahasiswa	13213065
Program Studi	Teknik Elektro ITB
Latar Belakang	
Tempat lahir	Jakarta
Tanggal lahir	02-11-1995
Alamat (libur)	Komplek Timah Pondok Labu D2, Jakarta
Pendidikan	SMA Negeri 34 Jakarta (2010-2013) Institut Teknologi Bandung (2013-sekarang)
Estimasi kapasitas kerja	
Alamat (masa kuliah)	Jalan Dago Asri IV no.20, Bandung
Status tempat tinggal	Kost
Mobilitas (skala 5)	4
Keahlian	Programming : C, C#, Matlab, VHDL, Software : Microsoft Office, Altium Designer, LT Spice, Visual Studio, Autodesk Inventor, Quartus, Proteus Professional
Jumlah kredit yang diambil untuk Semester I = 2016/2017 di luar EL4096	
Perkiraan alokasi waktu untuk seluruh kegiatan non-akademis (per minggu)	
Perkiraan alokasi waktu untuk seluruh kegiatan non-akademis (per minggu)	



INSTITUT TEKNOLOGI BANDUNG

PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 **(022)2508135-36**, **(022)2500940**
BANDUNG 40132

Dokumentasi Produk Tugas Akhir

Lembar Sampul Dokumen

Judul Dokumen

TUGAS AKHIR TEKNIK ELEKTRO:
*Sistem Jaringan Detektor Gempa dan Tsunami
Decision Support System*

Jenis Dokumen

SPESIFIKASI

Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB

Nomor Dokumen

B200-01-TA1617.01.069

Nomor Revisi

01

Nama File

B200-01-TA1617.01.069.docx

Tanggal Penerbitan

12 May 2017

Unit Penerbit

Prodi Teknik Elektro - ITB

Jumlah Halaman

17

(termasuk lembar sampul ini)

Data Pemeriksaan dan Persetujuan

Ditulis Oleh	Nama Tanggal Nama Tanggal Nama Tanggal	Christoporus Deo Putratama 25 September 2016 Bramantio Yuwono 25 September 2016 Kevin Shidqi Prakoso 25 September 2016	Jabatan Tanda Tangan Jabatan Tanda Tangan Jabatan Tanda Tangan	Anggota Kelompok Anggota Kelompok Anggota Kelompok
Diperiksa Oleh	Nama Tanggal	Dr. techn. Ary Setijadi Prihatmanto 25 September 2016	Jabatan Tanda Tangan	Dosen Pembimbing
Disetujui Oleh	Nama Tanggal	Dr. techn. Ary Setijadi Prihatmanto 25 September 2016	Jabatan Tanda Tangan	Dosen Pembimbing

DAFTAR ISI

1 PENGANTAR	4
1.1 RINGKASAN ISI DOKUMEN	4
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN	4
1.3 REFERENSI	4
1.4 DAFTAR SINGKATAN	5
2 SPESIFIKASI.....	5
2.1 DEFINISI, FUNGSI DAN SPESIFIKASI DARI SOLUSI	5
2.2 SPESIFIKASI TUGAS AKHIR	6
2.3 PENJELASAN FUNGSI, FEATURE, DAN VERIFIKASI	7
2.3.1 <i>Penjelasan Fungsi dan Fitur.</i>	7
2.3.2 <i>Penjelasan Verifikasi</i>	8
2.4 DESIGN	9
2.5 BIAYA DAN JADWAL	12
JADWAL PENGERJAAN	13

Catatan Sejarah Perbaikan Dokumen

VERSI	TANGGAL	OLEH	PERBAIKAN
1	25 September 2016	C. Deo P. Bramantio Y. Kevin S.P.	Versi pertama dokumen B200
2	1 Mei 2017	C. Deo P. Bramantio Y. Kevin S.P.	Revisi pertama dokumen B200

Proposal Proyek Pengembangan Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*

1 Pengantar

1.1 RINGKASAN ISI DOKUMEN

Isi Dokumen B200 ini merupakan salah satu kelengkapan dokumen proyek pengembangan Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*. Pada dokumen ini dijelaskan mengenai spesifikasi produk yang akan dikembangkan dan rencana pengujian yang akan diterapkan. Pembahasan mengenai spesifikasi produk meliputi definisi, fungsi, dan standar produk. Sedangkan rencana pengujian hasil desain dijabarkan dalam pengujian kelengkapan, keandalan, dan perawatan.

1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen

Tujuan dari penulisan dokumen B200 ini adalah:

- a. sebagai acuan spesifikasi pada pengerjaan proyek tugas akhir Pengembangan Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*, dan
- b. sebagai dokumentasi pengerjaan proyek tugas akhir Pengembangan Sistem Jaringan Detektor Gempa dan Tsunami *Decision Support System*.

Dokumen B200 ini ditujukan untuk tim dosen pembimbing TA1617.01.069 dan tim tugas akhir Teknik Elektro ITB 2016

1.3 REFERENSI

- [1] <http://www.codeproject.com/Articles/1117162/Detecting-Seismic-Waves-with-Android-and-IOT>, Diakses pada 18 September 2016.
- [2] Taufique Z. Redhwan, "A Neyman-Pearson approach to the development of low cost earthquake detection and damage mitigation system using sensor fusion", Electronics, Circuits and Systems (ICECS), 2014 21st IEEE International Conference on, 2014.
- [3] Dominguez, L.A., Yildirim, B., Husker, A.L., Cochran, E.S., Christensen, C.M., Cruz-Atienza, V.M., and Lawrence, J.F., The Red Atrapa Sismos (Quake Catcher Network in Mexico): Assessing performance during large and damaging earthquakes, *Seis. Res. Letts.*, 86, 848-855, 2015.
- [4] Yildirim, B., Cochran, E.S., Chung, A.I., Christensen, C.M., and Lawrence, J.F., On the reliability of Quake Catcher Network earthquake detections, *Seis. Res. Letts.*, 86, 856-869, 2015.
- [5] Neighbors, C., Liao, E.J., Cochran, E.S., Funning, G.J., Chung, A.I., Lawrence, J.F., Christensen, C., Miller, M., Belmonte, A., and Sepulveda, H.H.A., Investigation of the high-frequency attenuation parameter, kappa, from aftershocks of the 2010 Mw8.8 Maule, Chile earthquake, *Geophys. J. Int.*, 200, 200-215, doi: 10.1093/gji/ggu390, 2015.
- [6] Lawrence, J.F., Cochran, E.S., Chung, A., Kaiser, A., Christensen, C.M., Allen, R., Baker, J.W., Fry, B., Heaton, T., Kilb, D., Kohler, M.D., Taufer, M., Rapid Earthquake Characterization Using MEMS Accelerometers and Volunteer Hosts Following the M 7.2 Darfield, New Zealand, Earthquake, *Bull. Seis. Soc. Am.*, 104, 184-192, 2014.

- [7] Evans, J.R., Allen, R.M., Chung, A.I., Cochran, E.S., Guy, R., Hellweg, M., Lawrence, J.F., Performance of Several Low-Cost Accelerometers, *Seism. Res. Letts.*, 85, 147-158, 2014.
- [8] Benson, K., Schlachter, S., Estrada, T., Taufer, M., Lawrence, J., Cochran, E., On the powerful use of simulations in the Quake-Catcher Network to efficiently position low-cost earthquake sensors, *Future Generation Computer Systems*, 29, 2128-2142, 2013.

1.4 DAFTAR SINGKATAN

SINGKATAN	ARTI
APBN	Anggaran Pendapatan dan Belanja Negara
BMKG	Badan Meteorologi, Klimatologi, dan Geofisika
BPPT	Badan Pengkajian dan Penerapan Teknologi
DSS	Decision Support System
GPS	Global Positioning System
GITEWS	German Indonesian Tsunami Early Warning System
InaTEWS	Indonesia Tsunami Early Warning System
ITB	Institut Teknologi Bandung
LAPAN	Lembaga Penerbangan dan Antariksa Nasional
LIPI	Lembaga Ilmu Pengetahuan Indonesia
Pemda	Pemerintah daerah
PPMB	Pusat Penelitian Mitigasi Bencana
PPTIK	Pusat Penelitian Teknologi, Informasi dan Komunikasi

2 SPESIFIKASI

2.1 Definisi, Fungsi dan Spesifikasi dari Solusi

Sistem Jaringan Detektor Gempa dan Tsunami merupakan sistem multikomponen yang terintegrasi dengan jaringan nirkabel. Secara garis besar sistem ini terbagi menjadi tiga komponen, yaitu pengambilan data seismik yang didapatkan dari seismometer BMKG, Website USGS, dan Quake Catcher Network yang diintegrasikan dengan sensor yang dibuat, data center tempat untuk mengolah dan menggabungkan data yang diperoleh serta Tsunami Disaster Management yang bertugas dalam hal diseminasi dan evakuasi ketika terjadi gempa dan tsunami.

Seismometer BMKG terdiri dari jaringan seismik yang merupakan bagian dari InaTEWS, atau Indonesia Tsunami Early Warning System. Sistem ini mengandung 160 broadband seismometer yang disebar di seluruh wilayah Indonesia. Seismometer tersebut ada yang sepenuhnya dimiliki oleh Indonesia, namun ada juga yang dimiliki negara asing seperti Jerman, Republik Rakyat Tiongkok, dan Amerika Serikat. Seismometer ini diharapkan mampu mengirim data seismik melalui internet ke pusat pemrosesan data kami. Sistem seismik juga terdiri dari *buoy* yang disebar di perairan Indonesia, terutama di dekat sesar patahan yang rawan terjadi gempa dan tsunami.

USGS (*United States Geological Survey*) adalah lembaga ilmiah di Amerika Serikat yang mempelajari keadaan alam di Amerika Serikat, sumber daya alam yang terdapat di sana, serta fenomena yang mungkin mengancamnya. USGS memiliki seismometer dan sensor gempa lainnya yang tersebar di seluruh wilayah Amerika Serikat. USGS juga memiliki

sensor yang tersebar di seluruh dunia, termasuk di Indonesia. USGS menyediakan data gempa terkini melalui situs USGS, termasuk di dalamnya data seperti lokasi (bujur dan lintang) dan kedalaman, kekuatan, serta waktu terjadinya gempa. Data ini akan digunakan juga untuk membantu analisis yang dilakukan.

Diharapkan sistem yang dibuat ini dapat memenuhi beberapa kriteria, sebagai berikut

- **Harga Sensor yang dibuat Lebih Murah dibandingkan Seismometer**

Sensor yang digunakan harus memiliki harga yang lebih murah jika dibandingkan dengan seismometer. Dengan harga sensor yang lebih murah, dapat digunakan sensor pembacaan data seismik dari suatu tempat yang banyak diimplementasikan di banyak tempat yang akan membuat pembacaan menjadi lebih akurat. Dibandingkan dengan seismometer BMKG yang harganya diatas 300 juta rupiah, sensor kami ditargetkan hanya seharga 3 juta rupiah. Dengan begitu, dapat dibuat ratusan sensor yang bisa disebar di sepanjang pesisir barat Sumatera dan pesisir selatan Jawa, 2 garis pantai yang paling rawan terjadi tsunami.

- **Dapat Mengirimkan Data secara Real-Time dan Akurat**

Ketika terjadi gempa dan tsunami diharapkan sistem ini dapat memberikan respon dengan cepat dan akurat. Oleh karena itu dibutuhkan sistem yang dapat membaca serta mengolah data secara Real-Time dan Akurat agar peringatan dini yang dilakukan sistem ini dapat berjalan dengan efektif dan efisien.

- **Dapat Memberikan Peringatan Dini yang Efektif dan Efisien**

Agar dapat membantu korban ketika terjadi gempa dan tsunami, diharapkan sistem ini agar dapat menginformasikan peringatan dini kepada masyarakat yang berada di lokasi gempa dan tsunami dengan efektif dan efisien.

2.2 Spesifikasi Tugas Akhir

Pada tugas akhir ini, produk yang dibuat dibatasi fungsinya, yaitu hanya sebagai sistem pembacaan data seismik serta pengiriman data tersebut ke data center. Pembatasan ini berasalan karena produk yang dibuat ini merupakan bagian dari suatu proyek riset yang dilakukan oleh PPTIK ITB, sehingga pada bagian pengolahan data dan Tsunami Disaster Management sudah ada yang menangani. Data seismik yang dikirim ke data center diperoleh dari tiga sumber, yaitu Seismometer BMKG, Website USGS, dan Quake Catcher Network yang diintegrasikan dengan sensor yang kita buat.

Data yang diperoleh dari Seismometer BMKG dan Website USGS dapat diperoleh dengan menggunakan beberapa software seperti Seiscomp dan Comcat. Sedangkan data seismik untuk aplikasi Quake Catcher network menggunakan sensor yang kita buat. Sistem pembacaan dan pengiriman data seismik ini harus memenuhi beberapa spesifikasi agar membuat sistem jaringan detektor gempa dan tsunami decision support system dapat berjalan dengan efektif dan efisien.

- Dapat Mengirim Data secara Real-Time dan Akurat
- Dapat Mengirim Data dengan Menggunakan Jaringan Nirkabel

- Memiliki Safe Mode dan Backup Power pada Sensor sehingga ketika terjadi gempa dan tsunami Sensor dapat Tetap Berfungsi dengan baik
- Harga Sensor Lebih Murah dari Harga Seismometer
- Dapat Mendeteksi Gempa dengan Magnitude >4 pada Jarak 100 km

2.3 Penjelasan fungsi, feature, dan verifikasi

2.3.1 Penjelasan Fungsi dan Fitur

Sistem Jaringan Detektor Gempa dan Tsunami yang dibuat memiliki beberapa fitur, yaitu sebagai berikut.

- Pembacaan Data Seismik secara Akurat

Agar proses diseminasi dapat berjalan efektif dan efisien, pembacaan sensor harus akurat. Pembacaan sensor harus memiliki error pembacaan yang kecil sehingga error tersebut tidak mengganggu pengolahan dan penggabungan data pada data center. Pengujian fungsi ini dilakukan dengan menggunakan gempa buatan yang membuat pembacaan sensor mengindikasikan adanya getaran dan perubahan kecepatan sudut. Dengan referensi data yang diperoleh dari gempa buatan tersebut, data yang diperoleh saat testing akan dibandingkan dengan data referensi sehingga fungsi ini dapat diverifikasi peformanya.

- Pengiriman Data Seismik secara Real-Time dengan Komunikasi Nirkabel

Agar sistem ini dapat memberi peringatan ketika gempa dan tsunami terjadi, diperlukan adanya mekanisme pengiriman data seismik secara Real-Time dengan menggunakan Komunikasi Nirkabel. Untuk memverifikasi fungsi ini, testing dapat dilakukan dengan cara menghitung waktu update data seismik yang diperoleh dari Seismometer BMKG, Website USGS, dan Quake Catcher Network yang diintegrasikan dengan sensor yang dibuat pada data center. Kebutuhan peformansi dari fungsi ini terpenuhi jika waktu yang dibutuhkan untuk mengirim data kurang dari jumlah waktu yang dibutuhkan untuk melakukan analisis & penggabungan data dan golden time (5 menit).

- Analisis Data Seismik

Setelah data seismik diterima oleh pusat data, perlu dilakukan analisis dari data tersebut agar bisa didapatkan kesimpulan yang bermanfaat. Perlu metode agar bisa membedakan getaran yang hanya dirasakan oleh satu sensor, yang kemungkinan besar berupa *noise*, dan gempa sungguhan. Apabila memang terjadi gempa sungguhan, perlu juga analisis mengenai karakteristik gempa, seperti lokasi, kedalaman, serta kekuatan dari gempa tersebut.

- Sistem Peringatan

Apabila ada gempa terjadi dan hasil analisis menunjukkan bahwa gempa tersebut mempunyai potensi merusak dan/atau berpotensi tsunami, maka akan ada sistem peringatan berupa alarm atau semacamnya yang bisa memberitahu orang-orang pada saat itu bahwa ada gempa terjadi dan detil dari gempa tersebut.

Sistem ini dapat didemonstrasikan dengan cara menunjukkan data yang dibaca dari ketiga sumber yang telah disebutkan secara bersamaan. Lalu ketika dibuat gempa buatan untuk sensor, data yang diterima pada data center haruslah mengindikasikan bahwa gempa akan terjadi.

2.3.2 Penjelasan Verifikasi

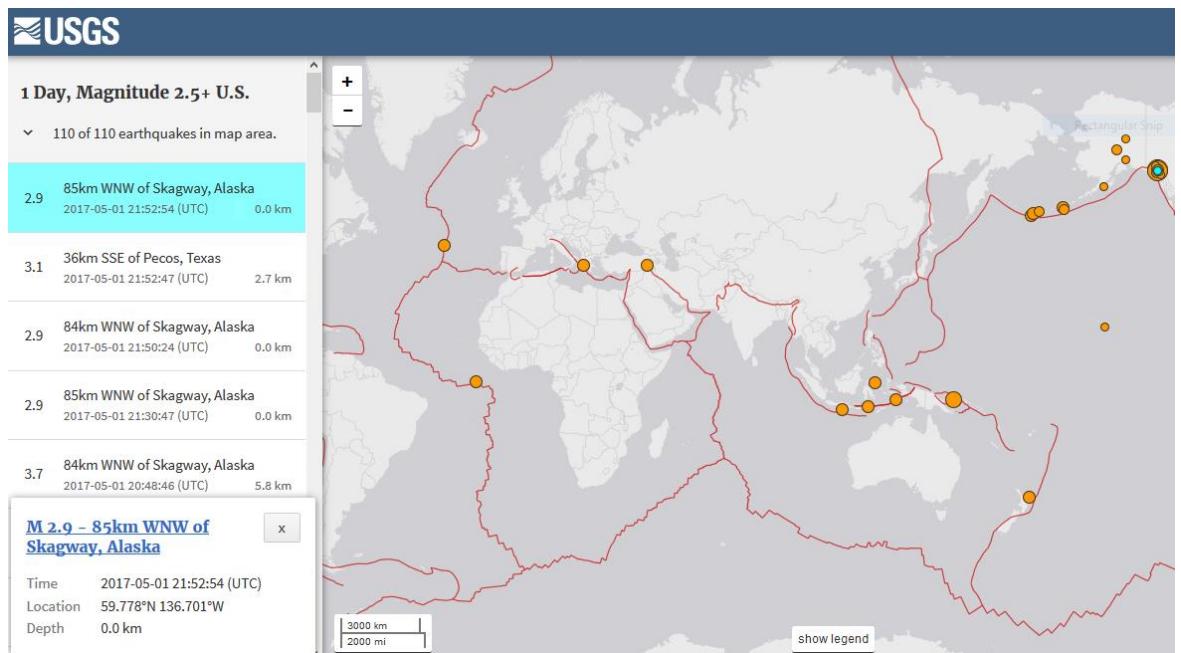
- Verifikasi Sensor

Verifikasi dari sensor gempa yang dibuat direncanakan sebagai berikut. Pertama, sensor gempa akan dikalibrasi dengan membandingkannya dengan seismometer sungguhan yang telah dikalibrasi sebelumnya. Kedua sensor harus menunjukkan data yang hampir sama, terutama apabila ada getaran buatan yang ditimbulkan di sekitar sensor tersebut. Seismometer yang akan digunakan ada di LIPI di Bandung. Sensor juga akan dibiarkan menyala untuk beberapa lama bersebelahan dengan seismometer, lalu data dari kedua sensor akan dibandingkan. Selanjutnya, kalibrasi pada sensor akan dilakukan apabila dibutuhkan.

Kedua, sensor akan diuji dengan menggunakan *shake table*. Shake table adalah alat yang mempunyai permukaan yang bisa digetarkan dengan percepatan tertentu. Sensor akan diletakkan di permukaan tersebut, dan getaran yang terbaca pada sensor berupa percepatan haruslah sama dengan pengaturan *shake table*. Shake table yang akan digunakan ada di Laboratorium Dinamika PAU ITB.

- Verifikasi Algoritma

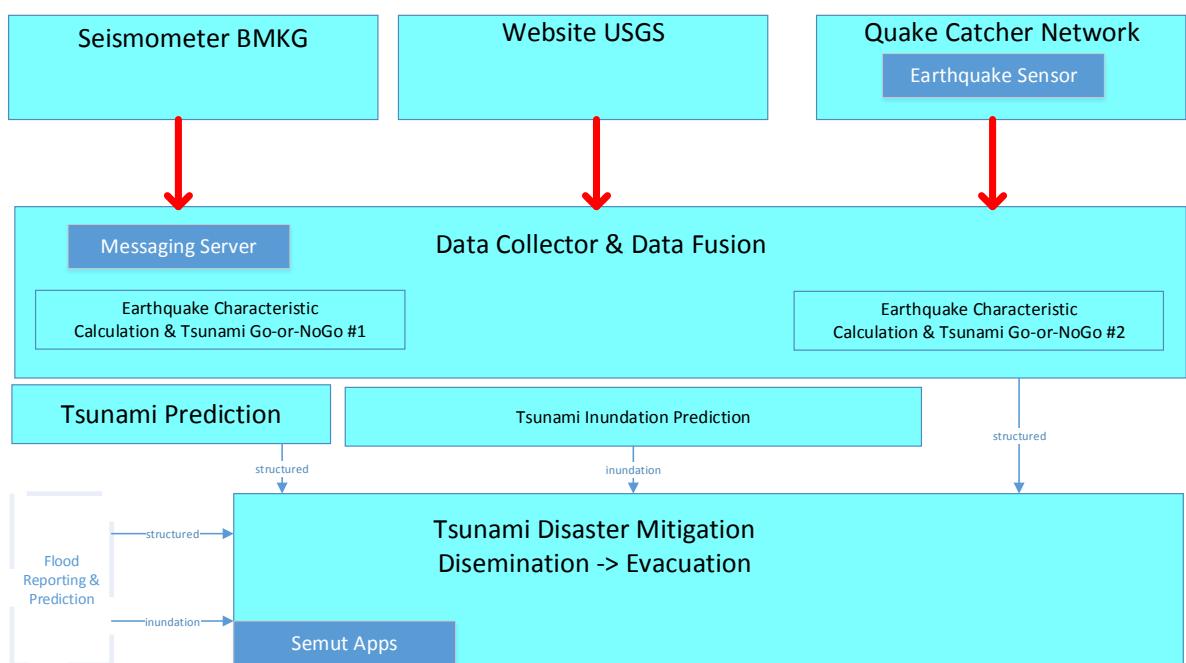
Karena gempa sungguhan yang mempunyai kekuatan lebih dari 4 jarang terjadi dan tidak bisa diprediksi, kami menggunakan data gempa yang telah berlalu untuk menguji algoritma kami. Kami akan meminta akses data seismometer, baik itu seismometer milik BMKG ataupun USGS. Setelah itu, kami melihat data gempa bulan terakhir yang terjadi di sekitar lokasi-lokasi dari seismometer. Lalu data gempa akan dicocokkan dengan pembacaan seismometer. Algoritma pemrosesan akan diterapkan pada data seismometer tepat saat terjadi gempa. Lalu, kesimpulan dari algoritma tersebut akan dibandingkan dengan kesimpulan yang dicapai oleh USGS atau BMKG. Kesimpulan yang dicapai harus akurat untuk gempa diatas magnitude 4.



Gambar 1 Website USGS untuk Verifikasi

2.4 Design

Dalam membuat desain sistem yang akan dirancang, perlu diketahui cakupan dari sistem melalui skema sistem yang akan dijelaskan pada gambar di bawah ini.



Gambar 2 Diagram Sistem

Penjelasan diagram blok:

Seismometer BMKG digunakan untuk *monitoring* gelombang seismik di seluruh Indonesia, dan disediakan aksesnya oleh BMKG. Seismometer ini merupakan metode yang paling konvensional dalam pengawasan gempa, dan telah digunakan dalam sistem yang telah berjalan sekarang ini yaitu InaTEWS. Seismometer ini merupakan salah satu sensor yang digunakan dalam sistem ini untuk mendeteksi gempa dan memprediksi tsunami. Seismometer juga bisa memberi data lokasi melalui GPS, sehingga bisa diprediksi bahaya tsunami atau tidak berdasarkan lokasi dan kedalaman gempa. Data tersebut dikirim melalui sebuah API *messaging*, melalui sebuah messaging server. Data tersebut akan dikirim ke server yang akan berfungsi sebagai pusat data dan tempat analisis data.

USGS (*United States Geological Survey*) adalah lembaga geologi Amerika Serikat yang menyediakan data *seismic waveform* melalui websitenya yang tersedia secara gratis. Data tersebut adalah data gempa yang diupdate secara *real-time* dan tersedia di situs <http://earthquake.usgs.gov/earthquakes/map>. Data ini digunakan sebagai sumber tambahan dalam deteksi gempa dan prediksi tsunami. Rincian data tersebut adalah sebagai berikut:

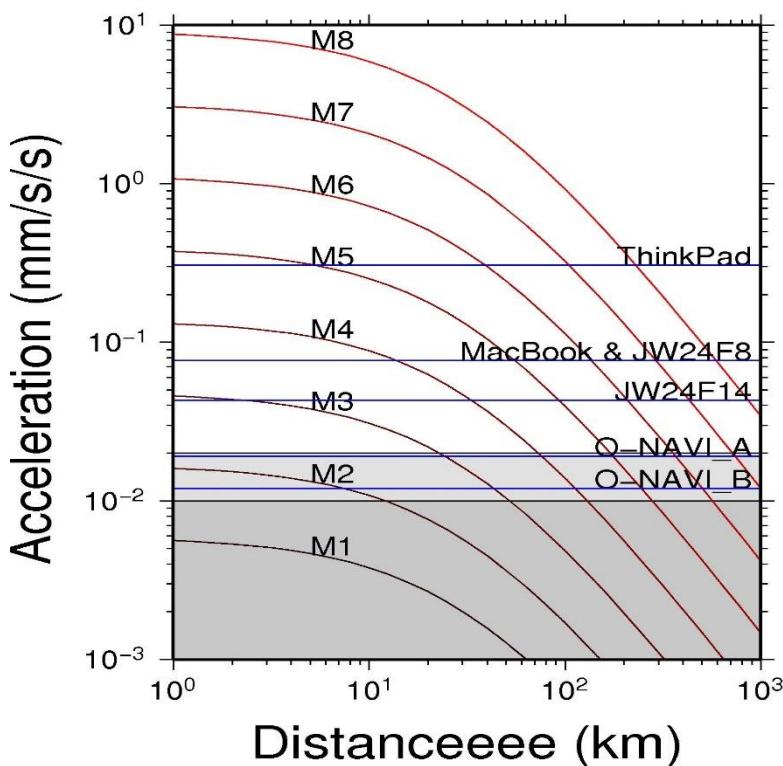
1. Kejadian gempa
2. Magnituda
3. Waktu dan tempat
4. Kedalaman gempa

Dari detail tersebut, bisa dibuat prediksi bahaya tsunami berdasarkan lokasi dan kedalaman gempa. Adanya data dari USGS yang mencakup seluruh dunia membantu kita dalam mendeteksi gempa yang agak jauh dari Indonesia sehingga tidak terdeteksi seismometer lokal, namun tetap bisa berpotensi tsunami yang membahayakan Indonesia.

QCN (Quake Catcher Network) adalah software *open-source* yang dikembangkan oleh Stanford University sebagai sistem detektor gempa berbasis sensor yang murah dan dapat dibuat oleh orang masyarakat umum. QCN diharapkan dapat menjadi sistem pendeteksi gelombang kuat terbesar di dunia yang murah dengan memanfaatkan sensor yang dihubungkan dengan komputer melalui USB, dan terhubung ke server QCN di California melalui internet. Jaringan QCN ini dapat memberitahukan suatu kejadian gempa melalui informasi yang didapat dari sensor-sensor yang dipasang ke jaringan tersebut.

Pada sistem ini, sensor yang akan dihubungkan ke QCN akan dibuat sendiri. Sensor tersebut berisi MEMS (Mini Electro-Mechanical Sensor), yaitu IMU (Inertial Measurement Unit) dan sensor piezoelectric. Sensor tersebut digunakan untuk mengukur getaran gempa dan orientasi dari getaran tersebut. Dengan cara ini dapat dibedakan antara gelombang primer dan sekunder dari gempa, sehingga kita bisa mengetahui kedalaman gempa tersebut. Melalui kerjasama banyak sensor di daerah-daerah tertentu, dapat juga ditentukan episentrum dari gempa melalui metode triangulasi, sehingga seperti data sebelumnya dapat ditentukan potensi tsunami dari gempa tersebut.

Sensor Noise



Gambar 3 Akurasi Sensor QCN

Dari gabungan ketiga data tersebut, akan dilakukan suatu proses analisis untuk menentukan bahaya atau tidaknya gempa tersebut, dan potensi tsunami. Analisis tersebut menggunakan model yang sudah dikembangkan berdasarkan kejadian gempa dan tsunami di masa lalu. Beberapa model yang akan kami gunakan adalah Tunami yang dikembangkan Tohoku University, Tunawi yang dikembangkan oleh Jerman, dan ComCod yang dikembangkan oleh USGS. Dengan adanya tiga data sumber, prediksi dapat dilakukan dengan lebih akurat. Melalui model ini, diharapkan bahwa analisis dapat dilakukan secepat mungkin, sehingga peringatan dini akan adanya tsunami dapat diberikan dalam waktu 5 menit dari sejak kejadian gempa.



Gambar 4 Sensor QCN

Prediksi akan adanya tsunami juga termasuk prediksi inundasi, yaitu prediksi seberapa jauh tsunami akan mencapai daratan. Hal tersebut dapat diprediksi dari kenyataan bahwa tinggi dari tsunami bergantung pada panjang gelombang serta kedalaman laut. Dengan menggunakan *Bathymetry*, yaitu pengukuran kedalaman dan topografi laut, dapat diprediksi tinggi tsunami yang akan terjadi. Bila ditambah dengan pengetahuan tentang topografi daratan, kita bisa memprediksi inundasi dari tsunami tersebut. Hasil dari proses analisis dan prediksi ini sangat penting karena akan menentukan bagaimana mitigasi bencana dilakukan.

Sistem diseminasi dan evakuasi digabung menjadi sistem mitigasi bencana, yang tidak terbatas pada gempa dan tsunami saja. Diseminasi adalah proses yang identik dengan alarm jika sudah terdeteksi gempa yang berpotensi tsunami. Diseminasi dapat dilakukan dengan memanfaatkan berbagai media seperti televisi, radio, HP, *speaker*, masjid, dan sebagainya. Proses diseminasi ini juga akan diintegrasikan dengan aplikasi Semut, yaitu Smart City & Intelligent Transportation System, sebuah app berbasis android yang sedang dikembangkan oleh LSKK ITB. Melalui app tersebut, pengguna bisa mendapatkan informasi mengenai peringatan dini gempa dan tsunami, jadi pengguna bisa mengetahui apakah dia perlu melakukan evakuasi dan jika ya, seberapa jauh. Selain itu, pengguna bisa menerima semacam *travel warning* jika ada daerah yang baru saja kena gempa, atau dalam risiko terkena gempa susulan

2.5 Biaya dan Jadwal

Uraikan dengan lebih terperinci Biaya dan Jadwal pelaksanaan real dari tugas akhir ini. Dalam bagian ini berikan uraian

- Analisis Biaya: Tenaga kerja dan beban kerja, Komponen dan bila diperlukan sumber dan outsourcing
- Jadwal: Tugas dinyatakan dengan cukup spesifik, Jadwal dengan tanggung jawab per orang yang jelas
- Sampaikan dengan jelas, apa yang akan selesai pada akhir TA1 (anggap sebagai suatu milestone dari proses design)

Biaya Sensor QCN

No	Komponen	Spesifikasi	Jumlah	Biaya
1	Main Board	Intel® Edison with Arduino Breakout Board	1	\$99.95
2	IMU	IMU Breakout - MPU-9250	1	\$14.95
3	Piezoelectric Sensor	Piezo Vibration Sensor	1	\$2.95
4	GPS Sensor	PAM-7Q GPS Module	1	\$49.99
5	Li-Ion Battery	Polymer Lithium Ion Battery - 2000mAh	1	\$12.95
6	Package		1	\$10.00
Total				\$190.79

Jadwal Penggerjaan

No	Deadline	Target Hands-on
1	23 September 2016	<ul style="list-style-type: none"> - Setup Ruangan Kerja - Koordinasi untuk peminjaman dan penggunaan peralatan lab yang diperlukan - Belajar menggunakan Oscilloscope HANTEK - Instalasi Arduino Studio 1.6.5, Visual Studio 2015, Android Studio
2	30 September 2016	<ul style="list-style-type: none"> - BMKG Seismometer data access - Opensource seismometer data processing software - Quake Catcher Network Software - Amica ESP8266, IMU Sensor & Interfacingnya - Hello World Visual Studio 2015 C#, Konsep Messaging Server
3	7 Oktober 2016	<ul style="list-style-type: none"> - Mempelajari protokol komunikasi <i>messaging server</i> - Akses data gempa USGS - Desain database gempa - Desain sensor QCN
4	14 Oktober 2016	<ul style="list-style-type: none"> - Purwarupa protokol komunikasi <i>messaging server</i> - Purwarupa Sistem USGS Earthquake downloader - Prototipng Sistem Sensor QCN & Akses server QCN - Desain Sistem Tsunami DSS
5	21 Oktober 2016	<ul style="list-style-type: none"> - Koneksi antara seismometer BMKG dengan database gempa - Koneksi Sistem USGS Earthquake downloader dengan database gempa - Purwarupa dan pengembangan Sistem Database Earthquake - Sensor QCN siap uji dan integrasi dengan database gempa - Purwarupa Sistem Tsunami DSS
6	28 Oktober 2016	<ul style="list-style-type: none"> - Pengolahan data seismometer BMKG - Pengolahan data dari website USGS - Pengolahan data dari server QCN - Pengujian software tsunami modelling
7	4 November 2016	<ul style="list-style-type: none"> - Purwarupa integrasi ketiga sumber data untuk penentuan bahaya tsunami - Desain integrasi dengan sistem mitigasi gempa - Desain integrasi dengan sistem mitigasi tsunami - Integrasi dengan model inundasi tsunami
8	11 November 2016	<ul style="list-style-type: none"> - Optimasi sistem penentuan bahaya tsunami - Optimasi sensor QCN - Desain integrasi dengan Semut Apps

9	18 November 2016	<ul style="list-style-type: none"> - Integrasi dengan Semut Apps - Pengujian sistem secara keseluruhan - Troubleshooting
10	25 November 2016	<ul style="list-style-type: none"> - Finalisasi sistem - Troubleshooting - Persiapan deployment dan produksi sensor - Prototyping Sistem Tsunami DSS

Kemudian, pembagian tugas sebagai berikut :

No	Tugas	Christoporus Deo	Kevin Shidqi	Bramantio
1	Desain Sensor			√
2	Desain GUI	√	√	
3	Manufaktur	√	√	√
4	Sensor dan Telemetri	√		√
5	Algoritma Pemrosesan		√	
6	Managerial		√	



INSTITUT TEKNOLOGI BANDUNG

PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 **(022)2508135-36, (022)2500940**
BANDUNG 40132

Dokumentasi Produk Tugas Akhir

Lembar Sampul Dokumen

Judul Dokumen	TUGAS AKHIR TEKNIK ELEKTRO: Sistem Detektor Gempa dan Tsunami Decision Support System	
Jenis Dokumen	PERANCANGAN	
	Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB	
Nomor Dokumen	B300-01-TA1617.01.069	
Nomor Revisi	01	
Nama File	B300	
Tanggal Penerbitan	12 May 2017	
Unit Penerbit	Prodi Teknik Elektro – ITB	
Jumlah Halaman	26	(termasuk lembar sampul ini)

Data Pemeriksaan dan Persetujuan				
Ditulis oleh	Nama	Kevin Shidqi	Jabatan	Anggota
	Tanggal	12 May 2017	Tanda Tangan	
	Nama	Bramantio Yuwono	Jabatan	Anggota
	Tanggal	12 May 2017	Tanda Tangan	
	Nama	Christoporos Deo	Jabatan	Anggota
		Putratama		
Diperiksa oleh	Tanggal	12 May 2017	Tanda Tangan	
	Nama	Dr. techn. Ary Setijadi	Jabatan	Dosen Pembimbing
	Tanggal	Prihatmanto		
		12 Mei 2017	Tanda Tangan	

Disetujui Oleh	Nama Tanggal	Dr. techn. Ary Setijadi Prihatmanto 12 Mei 2017	Jabatan	Dosen Pembimbing Tanda Tangan
-------------------	-----------------	-------------------------------------------------------	---------	-----------------------------------------

DAFTAR ISI

DAFTAR ISI.....	3
CATATAN SEJARAH PERBAIKAN DOKUMEN.....	3
PROPOSAL PROYEK SISTEM DETEKTOR GEMPA DAN TSUNAMI <i>DECISION SUPPORT SYSTEM</i>	5
1 PENGANTAR	5
1.1 RINGKASAN ISI DOKUMEN	5
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN	5
1.3 REFERENSI	5
1.4 DAFTAR SINGKATAN.....	5
2 PERANCANGAN	6
2.1 DESAIN SISTEM	6
2.2 DESAIN KOMUNIKASI	10
<i>2.2.1 Protokol Pengiriman Data dalam Jaringan</i>	10
2.3 DESAIN HARDWARE	12
<i>2.3.1 Sensor</i>	12
2.4 DESAIN SOFTWARE	20
<i>2.4.1 Desain Program Utama</i>	22
<i>2.4.2 Fungsi-fungsi Seiscomp3</i>	41
<i>2.4.3 Graphics User Interface untuk pengguna umum</i>	49
2.5 DESAIN BASIS DATA	51

Catatan Sejarah Perbaikan Dokumen

VERSI	TANGGAL	OLEH	PERBAIKAN
1	2 Desember 2017	C. Deo P. Bramantio Y. Kevin S.P.	Versi pertama dokumen B300

Proposal Proyek Sistem Detektor Gempa dan Tsunami *Decision Support System*

1 Pengantar

1.1 RINGKASAN ISI DOKUMEN

Dokumen ini merupakan kelanjutan dari dokumen B200. Dokumen ini berisi penjabaran perancangan dari proyek “Sistem Detektor Gempa dan Tsunami *Decision Support System*”. Kemudian dokumen ini terdiri dari 2 bab yaitu, pengantar dan perancangan.

Pada bab pengantar, terdiri dari ringkasan dari dokumen ini, tujuan penulisan dan kegunaan dokumen, referensi, dan daftar singkatan. Sedangkan pada bab perancangan berisi tentang desain sistem, desain komunikasi, desain hardware, desain software, desain pemodelan, dan desain basis data

1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen

Tujuan penulisan dokumen ini adalah sebagai berikut :

- Sebagai penjabaran perancangan dalam proyek “Sistem Detektor Gempa dan Tsunami *Decision Support System*”.
- Sebagai dokumentasi lanjutan pembuatan proyek “Sistem Detektor Gempa dan Tsunami *Decision Support System*”.

Dokumen ini ditujukan kepada dosen pembimbing tugas akhir dan tim tugas akhir Program Studi Teknik Elektro ITB sebagai bahan penilaian tugas akhir.

1.3 REFERENSI

- [1] Rudloff, Alexander, *German-Indonesian Tsunami Early Warning System (GITEWS) Decision Support System (DSS)*, Deutschen Zentrums für Luft- und Raumfahrt (DLR): Köln (2010)
- [2] <http://www.codeproject.com/Articles/1117162/Detecting-Seismic-Waves-with-Android-and-IOT>, Diakses pada 18 September 2016.
- [3] <https://www.rabbitmq.com/getstarted.html>, Diakses pada 18 September 2016.

1.4 DAFTAR SINGKATAN

SINGKATAN	ARTI
AMQP	Advanced Messaging Queueing Protocol
GeoJSON	Geo JavaScript Object Notation

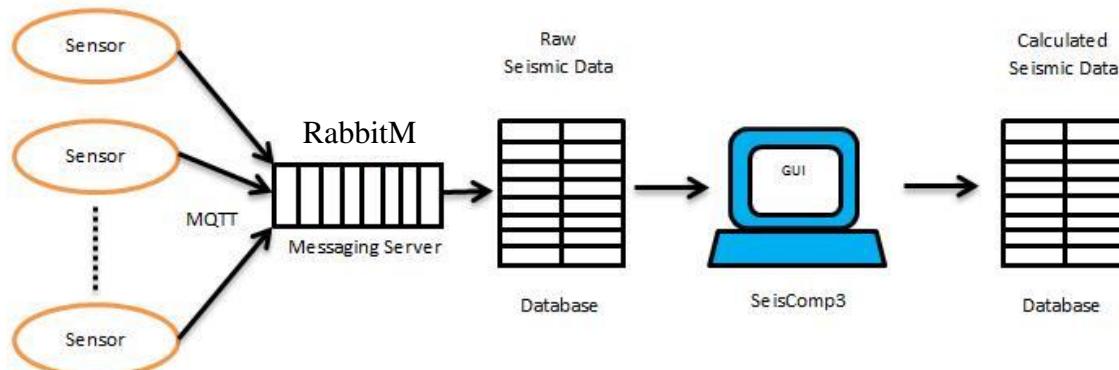
SINGKATAN	ARTI

2 PERANCANGAN

2.1 Desain Sistem

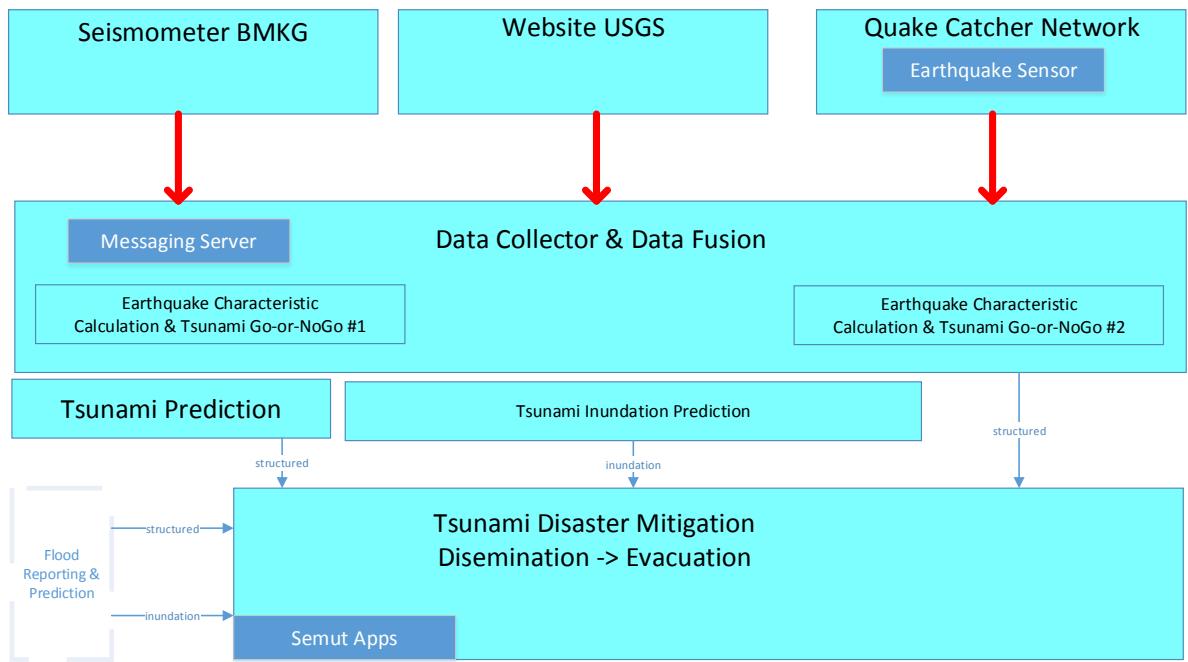
Earthquake Catcher Network adalah sistem pendekripsi gempa yang mengintegrasikan 3 sumber data berupa gelombang seismik, memroses data tersebut sehingga dihasilkan suatu keluaran yang berupa keputusan harus mengeluarkan suatu peringatan tsunami atau tidak. Sistem ini bekerja dengan cara mendekripsi suatu kejadian gempa, menentukan lokasi dan kedalaman pusat gempa, menentukan magnitudo dari kejadian gempa, lalu dengan sebuah model, memperkirakan bahaya tsunami yang mungkin ditimbulkan gempa tersebut. Selain itu, sistem juga memprediksi inundasi tsunami, atau seberapa jauh tsunami akan mencapai daratan.

Desain arsitektur dari sistem jaringan detektor gempa decision support system yang dirancang dapat dilihat pada gambar berikut ini.



Dapat dilihat dari beberapa sensor yang telah diletakkan, sensor tersebut akan mengirimkan beberapa data seperti sampling data percepatan tanah yang dibaca oleh sensor, lokasi sensor, dan timestamp ke messaging server (RabbitMQ) melalui protokol MQTT. Setelah itu GUI Server akan mengambil data mentah yang berada pada messaging server tersebut untuk disimpan pada database. Setelah data mentah tersebut tersimpan pada database, GUI Processing akan mengambil data tersebut dalam suatu interval waktu tertentu untuk diproses dan ditentukan apakah terjadi gempa dengan pembacaan beberapa sensor tersebut menggunakan aplikasi SeisComp3. Setelah proses dilakukan, data hasil

pemrosesan tersebut akan disimpan. Jika data hasil pemrosesan menunjukkan bahwa terjadi gempa maka GUI Processing tersebut akan memberikan peringatan pada GUI tersebut dan akan memberikan peringatan juga melalui twitter. Dalam membuat desain sistem yang akan dirancang, perlu diketahui cakupan dari sistem melalui skema sistem yang akan dijelaskan pada gambar di bawah ini.



Penjelasan diagram blok:

Seismometer BMKG digunakan untuk *monitoring* gelombang seismik di seluruh Indonesia, dan disediakan aksesnya oleh BMKG. Seismometer ini merupakan metode yang paling konvensional dalam pengawasan gempa, dan telah digunakan dalam sistem yang telah berjalan sekarang ini yaitu InaTEWS. Seismometer ini merupakan salah satu sensor yang digunakan dalam sistem ini untuk mendekripsi gempa dan memprediksi tsunami. Seismometer juga bisa memberi data lokasi melalui GPS, sehingga bisa diprediksi bahaya tsunami atau tidak berdasarkan lokasi dan kedalaman gempa. Data tersebut dikirim melalui sebuah API *messaging*, melalui sebuah messaging server. Data tersebut akan dikirim ke server yang akan berfungsi sebagai pusat data dan tempat analisis data.

USGS (*United States Geological Survey*) adalah lembaga geologi Amerika Serikat yang menyediakan data *seismic waveform* melalui websitenya yang tersedia secara gratis. Data tersebut adalah data gempa yang diupdate secara *real-time* dan tersedia di situs <http://earthquake.usgs.gov/earthquakes/map>. Data ini digunakan sebagai sumber tambahan dalam deteksi gempa dan prediksi tsunami. Rincian data tersebut adalah sebagai berikut:

1. Kejadian gempa

2. Magnitudo
3. Waktu dan tempat
4. Kedalaman gempa

Dari detail tersebut, bisa dibuat prediksi bahaya tsunami berdasarkan lokasi dan kedalaman gempa. Adanya data dari USGS yang mencakup seluruh dunia membantu kita dalam mendeteksi gempa yang agak jauh dari Indonesia sehingga tidak terdeteksi seismometer lokal, namun tetap bisa berpotensi tsunami yang membahayakan Indonesia.

QCN (Quake Catcher Network) adalah software *open-source* yang dikembangkan oleh Stanford University sebagai sistem detektor gempa berbasis sensor yang murah dan dapat dibuat oleh orang masyarakat umum. QCN diharapkan dapat menjadi sistem pendeteksi gelombang kuat terbesar di dunia yang murah dengan memanfaatkan sensor yang dihubungkan dengan komputer melalui USB, dan terhubung ke server QCN di California melalui internet. Jaringan QCN ini dapat memberitahukan suatu kejadian gempa melalui informasi yang didapat dari sensor-sensor yang dipasang ke jaringan tersebut.

Pada sistem ini, sensor yang akan dihubungkan ke QCN akan dibuat sendiri. Sensor tersebut berisi MEMS (Mini Electro-Mechanical Sensor), yaitu IMU (Inertial Measurement Unit) dan sensor piezoelectric. Sensor tersebut digunakan untuk mengukur getaran gempa dan orientasi dari getaran tersebut. Dengan cara ini dapat dibedakan antara gelombang primer dan sekunder dari gempa, sehingga kita bisa mengetahui kedalaman gempa tersebut. Melalui kerjasama banyak sensor di daerah-daerah tertentu, dapat juga ditentukan episentrum dari gempa melalui metode triangulasi, sehingga seperti data sebelumnya dapat ditentukan potensi tsunami dari gempa tersebut.

Dari gabungan ketiga data tersebut, akan dilakukan suatu proses analisis untuk menentukan bahaya atau tidaknya gempa tersebut, dan potensi tsunami. Analisis tersebut menggunakan model yang sudah dikembangkan berdasarkan kejadian gempa dan tsunami di masa lalu. Beberapa model yang akan kami gunakan adalah Tunami yang dikembangkan Tohoku University, Tunawi yang dikembangkan oleh Jerman, dan ComCod yang dikembangkan oleh USGS. Dengan adanya tiga data sumber, prediksi dapat dilakukan dengan lebih akurat. Melalui model ini, diharapkan bahwa analisis dapat dilakukan secepat mungkin, sehingga peringatan dini akan adanya tsunami dapat diberikan dalam waktu 5 menit dari sejak kejadian gempa.

Prediksi akan adanya tsunami juga termasuk prediksi inundasi, yaitu prediksi seberapa jauh tsunami akan mencapai daratan. Hal tersebut dapat diprediksi dari kenyataan bahwa tinggi dari tsunami bergantung pada panjang gelombang serta kedalaman laut. Dengan menggunakan *Bathymetry*, yaitu pengukuran kedalaman dan topografi laut, dapat diprediksi tinggi tsunami yang akan terjadi. Bila ditambah dengan pengetahuan tentang topografi daratan, kita bisa memprediksi inundasi dari tsunami tersebut. Hasil dari proses analisis dan prediksi ini sangat penting karena akan menentukan bagaimana mitigasi bencana dilakukan.

Sistem diseminasi dan evakuasi digabung menjadi sistem mitigasi bencana, yang tidak terbatas pada gempa dan tsunami saja. Diseminasi adalah proses yang identik dengan

alarm jika sudah terdeteksi gempa yang berpotensi tsunami. Diseminasi dapat dilakukan dengan memanfaatkan berbagai media seperti televisi, radio, HP, *speaker*, masjid, dan sebagainya. Proses diseminasi ini juga akan diintegrasikan dengan aplikasi Semut, yaitu Smart City & Intelligent Transportation System, sebuah app berbasis android yang sedang dikembangkan oleh LSKK ITB. Melalui app tersebut, pengguna bisa mendapatkan informasi mengenai peringatan dini gempa dan tsunami, jadi pengguna bisa mengetahui apakah dia perlu melakukan evakuasi dan jika ya, seberapa jauh. Selain itu, pengguna bisa menerima semacam *travel warning* jika ada daerah yang baru saja kena gempa, atau dalam risiko terkena gempa susulan.

Sistem Jaringan Detektor Gempa dan Tsunami yang dibuat memiliki beberapa fitur, yaitu sebagai berikut.

- **Pembacaan Data Seismik secara Akurat**

Agar proses diseminasi dapat berjalan efektif dan efisien, pembacaan sensor harus akurat. Pembacaan sensor harus memiliki error pembacaan yang kecil sehingga error tersebut tidak mengganggu pengolahan dan penggabungan data pada data center. Pengujian fungsi ini dilakukan dengan menggunakan gempa buatan yang membuat pembacaan sensor mengindikasikan adanya getaran dan perubahan kecepatan sudut. Dengan referensi data yang diperoleh dari gempa buatan tersebut, data yang diperoleh saat testing akan dibandingkan dengan data referensi sehingga fungsi ini dapat diverifikasi peformanya.

- **Pengiriman Data Seismik secara Real-Time dengan Komunikasi Nirkabel**

Agar sistem ini dapat memberi peringatan ketika gempa dan tsunami terjadi, diperlukan adanya mekanisme pengiriman data seismik secara Real-Time dengan menggunakan Komunikasi Nirkabel. Untuk memverifikasi fungsi ini, testing dapat dilakukan dengan cara menghitung waktu update data seismik yang diperoleh dari Seismometer BMKG, Website USGS, dan Quake Catcher Network yang diintegrasikan dengan sensor yang dibuat pada data center. Kebutuhan peformansi dari fungsi ini terpenuhi jika waktu yang dibutuhkan untuk mengirim data kurang dari jumlah waktu yang dibutuhkan untuk melakukan analisis & penggabungan data dan golden time (5 menit).

- **Safe Mode dan Backup Power**

Ketika gempa dan tsunami terjadi, sangat besar kemungkinan daya listrik dari jala-jala akan mati, sehingga sensor yang dibuat harus memiliki sumber daya cadangan dan Safe Mode agar sensor dapat menggunakan baterai cadangan. Fungsi ini berjalan baik jika ketika listrik jala-jala diputus, sensor akan secara automatis menggunakan baterai cadangan sebagai pensuplai daya ke sensor.

Sistem ini dapat didemonstrasikan dengan cara menunjukkan data yang dibaca dari ketiga sumber yang telah disebutkan secara bersamaan. Lalu ketika dibuat gempa buatan untuk

sensor, data yang diterima pada data center haruslah mengindikasikan bahwa gempa akan terjadi.

2.2 Desain Komunikasi

2.2.1 Protokol Pengiriman Data dalam Jaringan

Pada jaringan yang terdapat dalam Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System, standar data yang digunakan dalam pengiriman secara aktual adalah dalam bentuk sebuah pesan singkat yang memiliki isi yang padat dan jelas. Sebagai contoh, sebuah protokol pesan singkat untuk memberitakan terjadinya sebuah gempa dan/atau tsunami adalah sebagai berikut:

```
Tsunami Warning in BENGKULU, Eq Mag:7.0RS, 09-Dec-09 15:52:59 UTC,  
Loc:4.64S/101.11E,Dep:10km::BMKG
```

Dapat dilihat bahwa dalam mengirimkan pesan singkat maka diperlukan susunan aturan yang dapat dimengerti oleh sistem. Pada proyek ini, protokol komunikasi yang digunakan adalah berbasis Advanced Messaging Queueing Protocol (AMQP). Protokol ini dipilih karena mempunyai beberapa fitur yang sesuai dengan sistem yang digunakan. AMQP diterapkan oleh sebuah vendor komunikasi RabbitMQ sehingga akan digunakan RabbitMQ dengan metode AMQP sebagai protokol komunikasi sistem. RabbitMQ sendiri merupakan broker pesan, dimana sebuah pesan yang dikirimkan dari pengirim menuju tujuan akan difasilitasi oleh RabbitMQ. Protokol pesan dapat dimodifikasi sehingga:

- Pengiriman pesan dapat diurutkan dalam sebuah antrian baris;
- Melakukan penerbitan pesan ke seluruh bagian sistem jaringan;
- Melakukan langganan pesan oleh pengguna pada broker;
- Melakukan penghubungan dan penyaringan pesan;
- Seleksi pesan yang akan diperoleh;
- Kendali jarak jauh pada jaringan dengan metode *remote*.

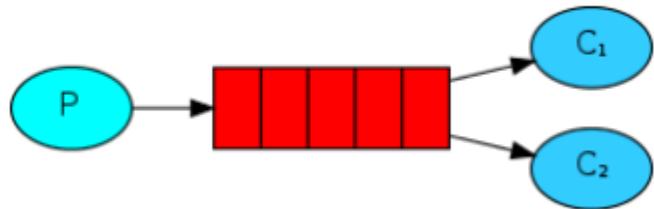
Fitur-fitur diatas dapat menguntungkan karena tidak semua data yang diperoleh merupakan informasi yang penting dan besar kemungkinan pada sistem bawa sebuah pesan dapat terkena distorsi akibat *bug* pada sistem. Selain itu mudah untuk mengendalikan sebuah sistem jaringan yang besar dengan protokol pesan AMQP ini.

Diagram blok dari sistem protokol komunikasi RabbitMQ adalah sebagai berikut:

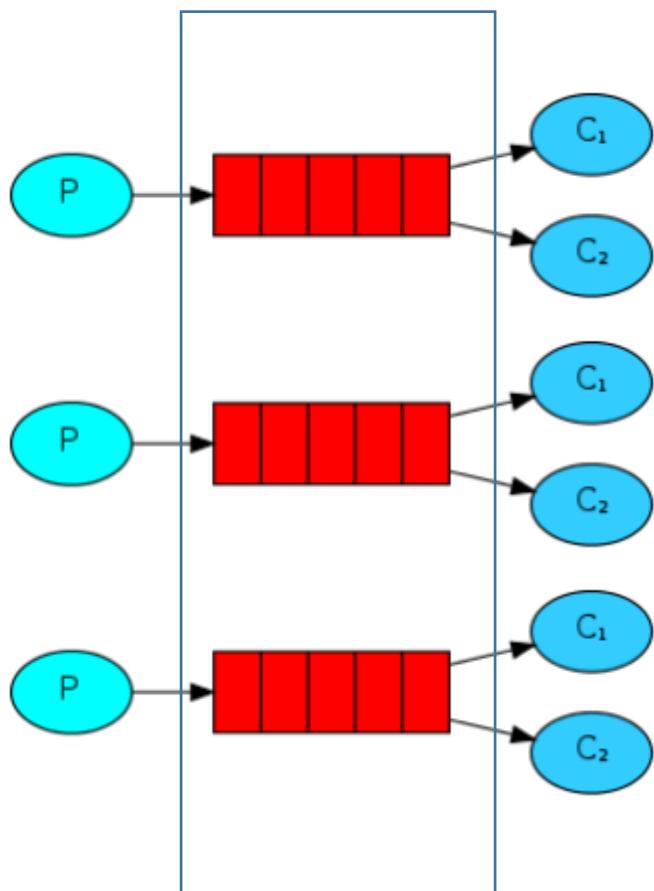


Gambar diagram blok diatas dari sistem paling sederhana dari RabbitMQ, dimana ada sebuah komponen sistem yang menghasilkan sebuah pesan, kemudian dimasukkan ke

dalam antrian dan selanjutnya dikirimkan kepada konsumen. Selanjutnya akan disertakan diagram blok dari fitur RabbitMQ berupa penerbitan pesan pada beberapa konsumen.



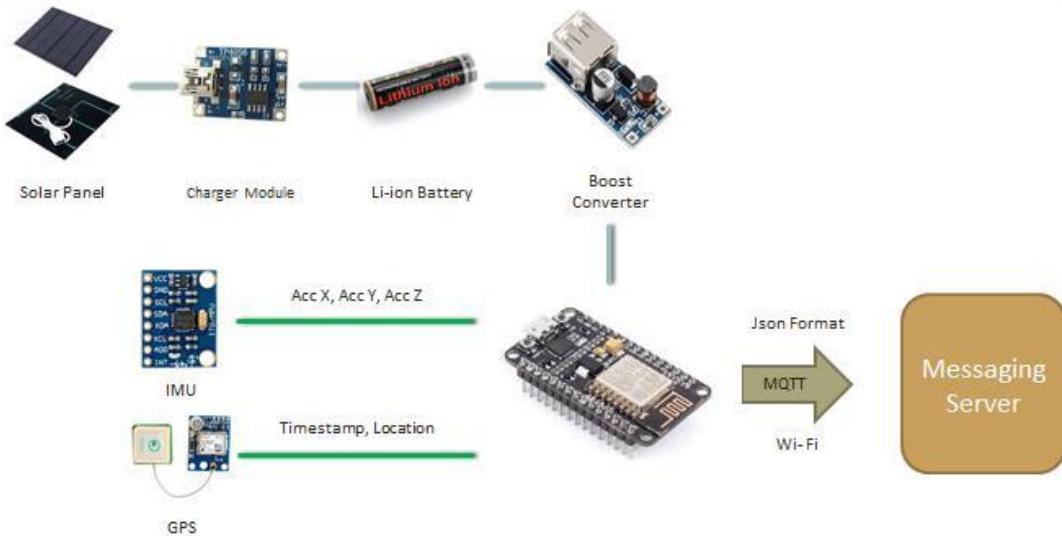
Pada Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System, p



2.3 Desain Hardware

2.3.1 Sensor

Pada sistem elektrikal sensor sendiri dibagi menjadi dua bagian, yaitu sistem daya dan sistem mikrokontroller. Untuk desain sistem elektrikal dapat dilihat pada diagram berikut ini.



Untuk menyuplai sensor yang harus bekerja selama 24 jam terus-menerus, digunakan solar panel yang akan menyuplai sensor pada siang hari. Karena solar panel tidak dapat menyuplai daya pada siang hari maka diperlukan rangkaian charging baterai, agar ketika malam sistem daya ini masih dapat menyuplai sistem mikrokontroler. Sehingga digunakan modul charge controller baterai Li-ion yang berbasis chip TP4056. Chip TP4056 ini dapat mengatur tingkat charging baterai Li-ion karena untuk mengisi baterai Li-ion dibutuhkan tegangan konstan pada 4.2 Volt dan arus yang berubah sesuai dengan kapasitas baterai yang sudah terisi. Baterai Li-ion dipilih karena sifatnya yang tidak mudah rusak jika dilakukan proses charge-discharge dibandingkan dengan baterai lain.

Untuk menentukan berapa daya solar panel dan kapasitas baterai yang dibutuhkan, dilakukan beberapa asumsi dalam perhitungan. Asumsi pertama adalah solar panel tidak akan bisa menyuplai daya selama 12 jam sehari yaitu dari jam 6 pagi sampai jam 6 malam. Asumsi kedua adalah solar panel hanya dapat menyuplai sensor selama 7 jam jika cuaca cerah. Dengan diketahui tegangan kerja sensor, daya maksimum yang dibutuhkan NodeMCU maka dapat diperoleh.

Pada malam hari :

$$V_{solar} = V_{battery} = V_{sensor} = 5 \text{ V}$$

$$I_{battery} = I_{sensor} = 1 \text{ A selama } 12 \text{ jam}$$

$$C_{battery} = 1 \text{ A} * 12 \text{ h} = 12000 \text{ mAh}$$

Pada siang hari :

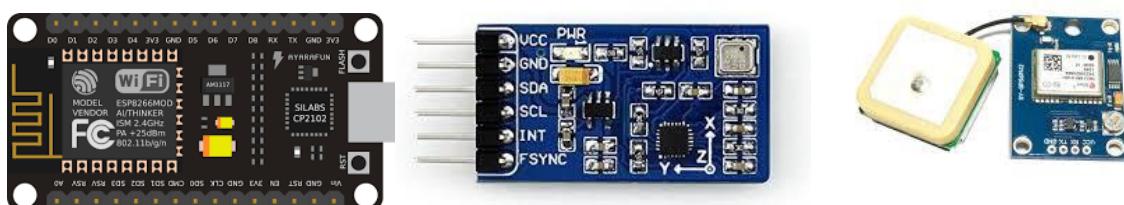
$$I_{solar} = I_{battery} + I_{sensor}$$

$$C_{battery} = 7 * I_{battery} = 12000 \text{ mAh}$$

$$I_{battery} = 1.7 \text{ A}, I_{solar} = 2.7 \text{ A}$$

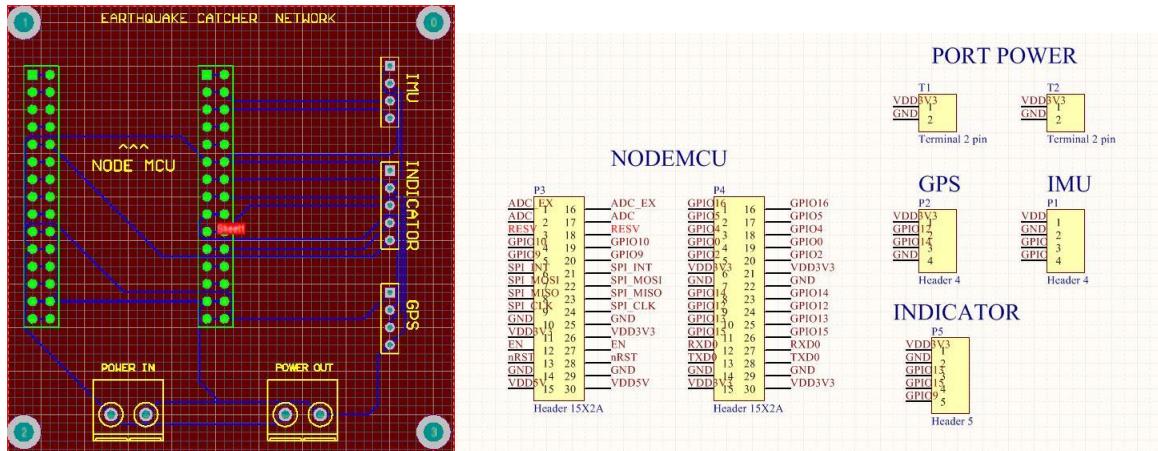
$$P_{solar} = 13.5 \text{ W}$$

Diperoleh bahwa dibutuhkan baterai dengan kapasitas 12000 mAh dan solar panel yang memiliki daya sebesar 13.5 W. Perhatikan bahwa perhitungan ini tidak memperhatikan fakta bahwa solar panel tidak akan menyuplai daya sebesar 13.5 W secara konstan dan loss daya pada rangkaian charge controller. Sehingga hitungan ini hanya sebagai dasar penentuan sistem daya yang tepat untuk sensor dan diperlukan percobaan lagi untuk mengetahui berapa daya solar panel yang dibutuhkan dan kapasitas baterai yang dibutuhkan.

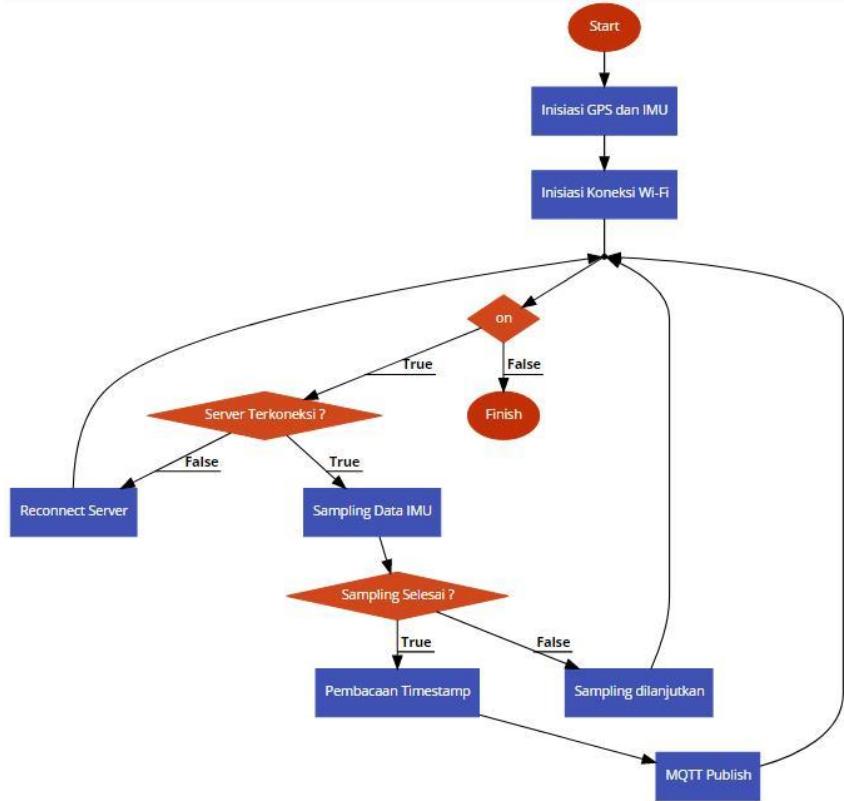


Pada sistem mikrokontroller, digunakan NodeMCU sebagai mikrokontroller dengan alasan harga yang murah dan sudah terdapat modul Wi-Fi pada NodeMCU. Untuk GPS digunakan modul GPS ublox Neo-6M dan untuk IMU digunakan MPU9255. Dipilih MPU9255 dibandingkan dengan IMU jenis lain berdasarkan pada adanya magnetometer yang dapat digunakan untuk mengoreksi simpangan antara sumbu x sensor dengan kutub utara.

Rangkaian dan desain PCB pada sistem mikrokontroler dapat dilihat pada diagram dibawah ini

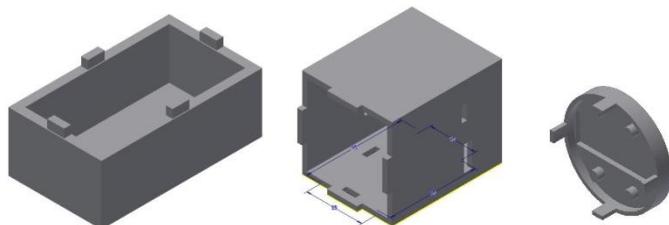


Dapat dilihat bahwa koneksi dari mikrokontroler NodeMCU ke GPS, IMU, dan modul Indikator digunakan header untuk memudah pemasangan dan instalasi. Ukuran board PCB yang didesain adalah 6.5 cm x 6.5 cm menyesuaikan dengan casing yang didesain. Untuk koneksi PCB ini ke sistem daya digunakan terminal dengan dua port sehingga lebih mudah dihubungkan antar sistem.



Sedangkan flowchart algoritma yang akan digunakan pada sensor dapat dilihat pada gambar diatas. Pada awal mula program dilakukan inisiasi untuk membuat modul GPS, IMU, dan koneksi Wi-Fi dapat digunakan. Pada kondisi menyala sensor akan melakukan pemeriksaan apakah sensor terhubung ke server, jika tidak sensor akan melakukan percobaan koneksi ke server ulang sampai sensor terkoneksi dengan server. Ketika sudah terkoneksi dengan server, sensor akan melakukan sampling data percepatan dengan periode 25 ms. Ketika telah diperoleh jumlah sampling yang diinginkan maka akan dilakukan proses publish message ke messaging server dengan melalui protokol MQTT. Tetapi sebelum dilakukan publish, dilakukan terlebih dahulu pembacaan Timestamp dari GPS. Setelah proses publish selesai dilakukan, sensor akan mengulangi proses yang sudah dijelaskan dari awal kembali.

Untuk menyatukan sistem daya dan sistem mikrokontroler yang sudah disebutkan tadi, diperlukan sebuah casing yang membuat sensor tidak mudah rusak atau terganggu dengan lingkungan luar. Berikut adalah desain 3D casing dengan menggunakan aplikasi Autodesk Inventor.





Gambar Casing 1 (Kiri Atas), Casing 2 (Tengah Atas), Casing 3 (Kanan Atas), Casing 4 (Kiri Bawah), Casing 5 (Tengah Bawah), Casing 6 (Kanan Bawah)

Desain pada gambar pertama berfungsi untuk tempat meletakkan baterai sehingga baterai mudah diganti dan dapat di tempel pada casing utama pada gambar kedua yang berfungsi sebagai tempat meletakkan PCB dan sistem daya. Pada casing utama tersebut, terdapat lubang untuk meletakkan casing baterai yang sudah dijelaskan sebelumnya dan lubang sebagai tempat meletakkan GPS dan sebagai tempat kabel IMU yang akan digunakan pada casing IMU pada gambar 3,4 dan 5. Gambar 3 merupakan tempat untuk meletakkan modul MPU9255 yang dapat ditutup oleh casing pada gambar 4 dan 5. Desain 3D pada gambar 6 berfungsi untuk menutup casing utama pada gambar 2 yang sudah dijelaskan sehingga kabel untuk solar panel dapat masuk ke modul sistem mikrokontroler dan casing utama. Casing pada gambar 6 juga berfungsi untuk meletakkan modul indikator.

Untuk alternatif desain terdapat beberapa pilihan, yaitu dari segi pemilihan mikrokontroler, sensor IMU yang digunakan, dan komunikasi ke server. Berikut adalah beberapa penjelasan alternatif desain.

2.3.1.1 Alternatif Desain Komponen Mikrokontroller & Mikrokomputer

Untuk melakukan fungsi sebagai sensor seismik, selain menggunakan NodeMCU, dapat digunakan mikrokomputer seperti Raspberry Pi 3 dan Intel Galileo Gen 2. Berikut adalah perbandingan spesifikasi antara NodeMCU, Raspberry Pi 3, dan Intel Galileo Gen 2.

NodeMCU

US\$2.85

160 MHz

64 KB for instruction and 96 KB for Data

4.5W



Raspberry Pi 3

US\$35

1.2 GHz quad-core ARM Cortex-A53

1 GB LPDDR2 RAM

6.7W

OS : Raspbian, Windows 10 IoT core, etc



Intel Galileo Gen 2

US\$74.95

400 MHz Intel Quark X1000

256 MB DDR3 RAM

15 W

OS: Linux, Windows IoT Core, etc



Dapat dilihat bahwa NodeMCU lebih unggul jika dibandingkan dengan Raspberry Pi 3 dan Intel Galileo Gen 2 dalam hal harga dan daya maksimum yang dikonsumsi. Sedangkan dapat dikatakan bahwa dari segi kekuatan pemrosesan, Intel Galileo Gen 2 dan Raspberry Pi 3 lebih baik jika dibandingkan dengan NodeMCU sehingga pemrosesan data seismik dapat dilakukan secara on-board, dilakukan pada masing-masing sensor. Dapat dikatakan jika pemrosesan dilakukan pada setiap sensor, maka sedikit berlebihan karena perlu cost lebih dlm komponen mikrokontroler. Sehingga dipilih NodeMCU karena harga yang jauh lebih murah dan daya yang dikonsumsi lebih rendah dibandingkan Raspberry Pi 3 dan Intel Galileo Gen 2.

2.3.1.2 Alternatif Desain Komponen IMU

Dalam proses desain, terdapat dua pilihan komponen sensor IMU yang ditemui. Perbandingan spesifikasi antara dua sensor IMU tersebut dapat dilihat sebagai berikut.

MPU9255

2.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-9255 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$ and integrated 16-bit ADCs
- Accelerometer normal operating current: $450\mu A$
- Low power accelerometer mode current: $8.4\mu A$ at $0.98Hz$, $19.8\mu A$ at $31.25Hz$
- Sleep mode current: $8\mu A$
- User-programmable interrupts
- Wake-on-motion interrupt for low power operation of applications processor
- Self-test

2.3 Magnetometer Features

The triple-axis MEMS magnetometer in MPU-9255 includes a wide range of features:

- 3-axis silicon monolithic Hall-effect magnetic sensor with magnetic concentrator
- Wide dynamic measurement range and high resolution with lower current consumption.
- Output data resolution of 14 bit ($0.6\mu T/LSB$) or 16 bit ($15\mu T/LSB$)
- Full scale measurement range is $\pm 4800\mu T$
- Magnetometer normal operating current: $280\mu A$ at $8Hz$ repetition rate
- Self-test function with internal magnetic source to confirm magnetic sensor operation on end products

2.4 Additional Features

The MPU-9255 includes the following additional features:

- Auxiliary master I^2C bus for reading data from external sensors (e.g. pressure sensor)
- $3.5mA$ operating current when all 9 motion sensing axes and the DMP are enabled
- VDD supply voltage range of $2.4 - 3.6V$
- VDDIO reference voltage for auxiliary I^2C devices
- Smallest and thinnest QFN package for portable devices: $3x3x1mm$
- Minimal cross-axis sensitivity between the accelerometer, gyroscope and magnetometer axes
- 512 byte FIFO buffer enables the applications processor to read the data in bursts
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- $10,000 g$ shock tolerant
- $400kHz$ Fast Mode I^2C for communicating with all registers
- $1MHz$ SPI serial interface for communicating with all registers

MPU6050

5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: $500\mu A$
- Low power accelerometer mode current: $10\mu A$ at $1.25Hz$, $20\mu A$ at $5Hz$, $60\mu A$ at $20Hz$, $110\mu A$ at $40Hz$
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I^2C bus for reading data from external sensors (e.g., magnetometer)
- $3.9mA$ operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of $2.375V - 3.46V$
- Flexible VLOGIC reference voltage supports multiple I^2C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: $4x4x0.9mm$
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- $10,000 g$ shock tolerant
- $400kHz$ Fast Mode I^2C for communicating with all registers
- $1MHz$ SPI serial interface for communicating with all registers (MPU-6000 only)
- $20MHz$ SPI serial interface for reading sensor and interrupt registers (MPU-6000 only)

Perbedaan MPU6050 dengan MPU9255 hanya terletak pada ada atau tidaknya magnetometer. MPU9255 memiliki magnetometer yang dapat mendeteksi simpangan antara sumbu x sensor IMU dengan kutub utara. Magnetometer diperlukan karena untuk mengoreksi simpangan saat sensor diletakkan agar semua sensor memiliki acuan x,y,z yang sama.

2.3.1.3 Alternatif Desain Komunikasi



SIM900 GSM GPRS Shield Moduue

Quad-Band 850 / 900/ 1800 / 1900 MHz

GPRS multi-slot class 10/8

GPRS mobile station class B

Compliant to GSM phase 2/2+

Class 4 (2 W @ 850 / 900 MHz)

Class 1 (1 W @ 1800 / 1900MHz)

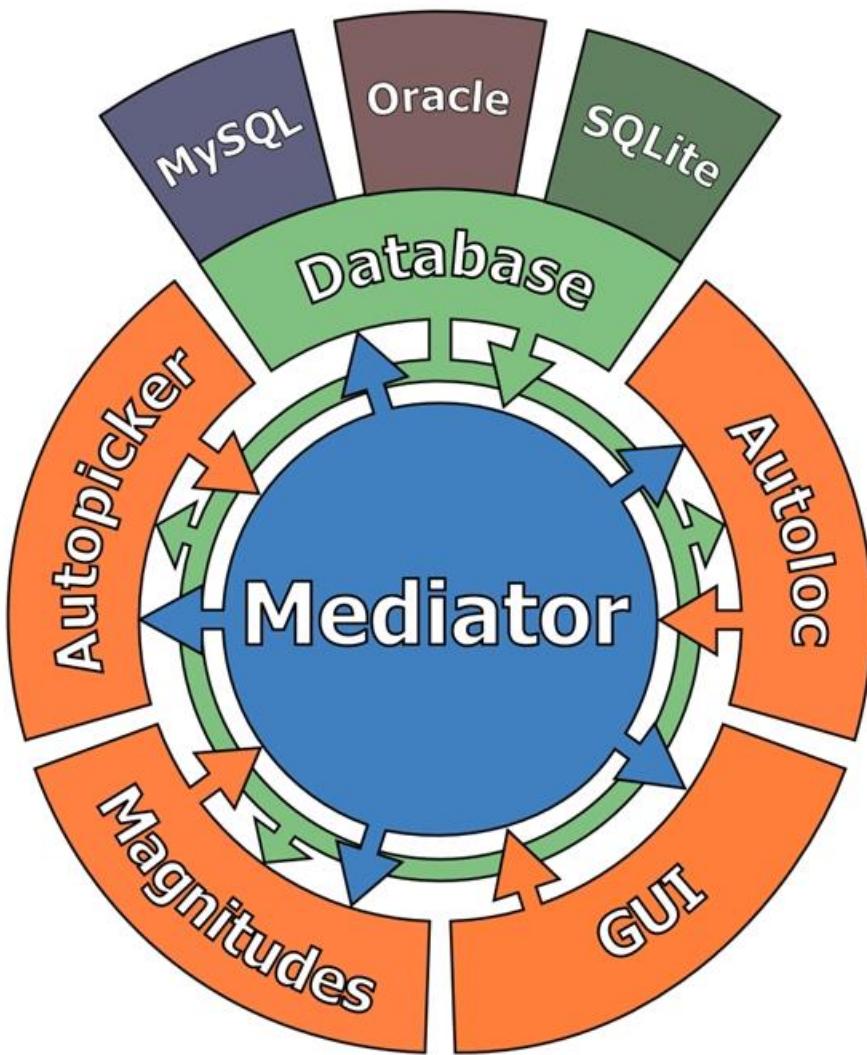
Pada bagian komunikasi kami memiliki alternatif desain dengan menggunakan komunikasi GSM jika dibandingkan dengan Wi-Fi. Wi-Fi memiliki band frekuensi dari 2.4, 3.6, 5, and 60 GHz sedangkan GSM memiliki band frekuensi dari 850 / 900/ 1800 / 1900 MHz. Dapat dilihat bahwa jaringan Wi-Fi memiliki kecepatan transfer data yang lebih cepat jika dibandingkan dengan GSM. Selain itu, hal yang perlu diperhatikan juga adalah masalah harga modul GSM yang cukup mahal , sekitar Rp285000.

2.4 Desain Software

Software untuk analisis data hasil perekaman sensor didasarkan pada program Seiscomp3, yaitu program analisis data seismik yang *open-source*. Seiscomp 3 dikembangkan oleh GEOFON di pusat riset geosains Jerman. Program ini didesain agar bisa menampilkan data seismik dari sensor, lokasi dari sensor pada suatu peta dunia, serta hasil analisis yang bisa didapatkan secara langsung dari data yang ada pada basis data. Hasil analisis itu berupa kekuatan gempa, lokasi gempa, spektrum frekuensi yang terekam, serta apakah getaran yang terekam oleh sensor merupakan gempa sungguhan atau hanya *noise* saja. Program ini dirancang agar bisa menggantikan helicorder secara mudah, murah, dan cepat.

Program ini akan ditulis dalam bahasa Java, agar mudah diintegrasikan dengan fungsi-fungsi yang ada pada Seiscomp3. Namun, Seiscomp3 hanya bisa berjalan pada sistem operasi Linux. Agar sistem bisa dijalankan pada semua sistem operasi, akan dibuat suatu program baru yang bisa dijalankan pada sistem operasi Windows, namun dengan menggunakan fungsi-fungsi yang ada pada Seiscomp3. Fungsi-fungsi yang digunakan akan dijelaskan pada bagian berikutnya.

Secara umum, gambaran software yang akan dikembangkan adalah sebagai berikut. Modul utama diberi nama mediator, yang akan menghubungkan semua bagian dari program. Mediator ini dibutuhkan karena fungsi-fungsi yang akan dipanggil ada dalam bentuk *library* yang indepenen, jadi akan lebih baik apabila fungsi-fungsi yang ada tersebut dipanggil secara terpisah apabila ada perintah dari user. Masing-masing fungsi tersebut melakukan tugas secara diskrit. Fungsi-fungsi yang akan dipanggil terutama adalah autoloc, autopicker, dan magnitude. Fungsi ini akan mengubah data seismik masing-masing menjadi data lokasi, data apakah terjadi gempa atau tidak, serta kekuatan dari gempa. Hasil dari fungsi ini akan ditampilkan pada GUI yang akan dikembangkan kemudian. Mediator juga menghubungkan sistem lain dengan basis data. Untuk sementara, jenis basis data yang didukung adalah MySQL, Oracle, dan SQLite. Pada perkembangan berikutnya, akan dibuat dukungan untuk jenis basis data lainnya seperti PostgreSQL.



Program ini dirancang agar bisa fleksibel. Sebagai fitur utama, program ini dirancang agar bisa terhubung dengan banyak jenis server gempa yang ada. Pada saat dokumen ini ditulis, program ini bisa terhubung dengan banyak jenis server, seperti *Earthworm*, IRIS DMC, SEED, dan SAC files. Untuk sensor kami sendiri, program ini juga dibuat agar bisa membaca data ASCII standar. Akses ke server ini dirancang agar bisa ditampilkan data baik dalam domain frekuensi maupun domain waktu di GUI. Selain itu, akan ada beberapa jenis tampilan yang disesuaikan dengan beberapa jenis pengguna. Untuk ilmuwan dan analis, disediakan tampilan yang detail dan fitur *zoom* yang memungkinkan analisis gelombang seismik dengan lebih akurat. Untuk seismolog amatir, disediakan tampilan yang lebih menyeluruh namun tetap mendetail. Untuk khayalak umum, disediakan tampilan minimalis yang dinamakan *kiosk*. Sasaran dari program ini adalah merangkul dan menyeluruh, sehingga sistem yang bisa dikonfigurasi adalah suatu prioritas.

Sistem otomatis SeisComP3 terdiri dari satu set aplikasi independen yang masing-masing melakukan tugas diskrit. Komunikasi antara aplikasi direalisasikan oleh sistem pesan

berbasis TCP / IP. Sistem pesan ini didasarkan pada toolkit open source "Spread" yang menyediakan layanan olahpesan berkinerja tinggi di jaringan area lokal dan luas. Di bagian atas "Spread" seorang mediator, disebut scmster yang menangani persyaratan tambahan dari SeisComP3 yang tidak disediakan secara native oleh "Spread". Sistem pesan digunakan untuk pertukaran meta data (misalnya picks) dan administrasi modul program. Model data SeisComP3 didasarkan pada skema QuakeML versi 0.5. QuakeML juga digunakan sebagai database object schema. Secara default SeisComP3 menggunakan database MySQL, tapi PostgreSQL juga didukung.

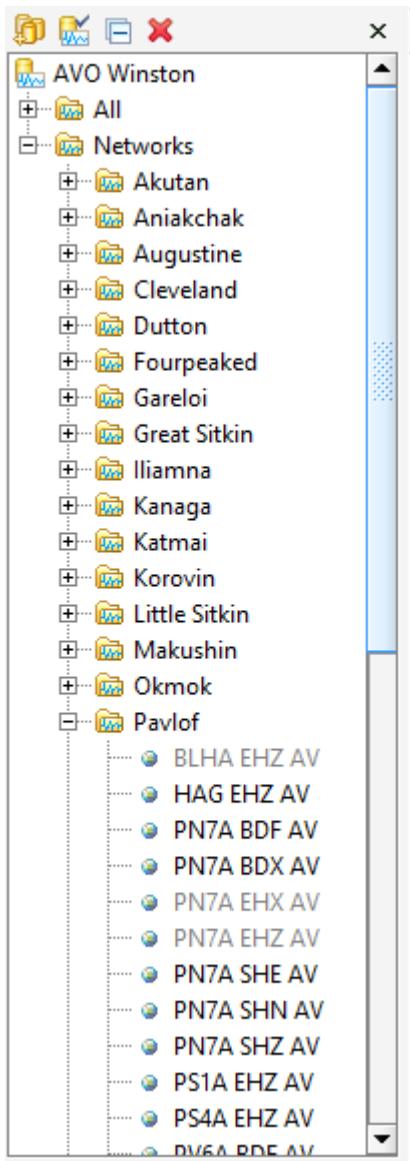
Akuisisi data waveform didasarkan pada protokol SeedLink yang mapan dan protokol ArcLink baru dikembangkan di GFZ Potsdam. Aplikasi di SeisComP3 dapat dibagi dalam empat kelompok yang berbeda: akuisisi data, pemrosesan, antarmuka pengguna grafis dan utilitas. Uraian rinci tentang aplikasi ada di bab 5. Tabel 4-1 menunjukkan gambaran singkat tentang aplikasi.

2.4.1 Desain Program Utama

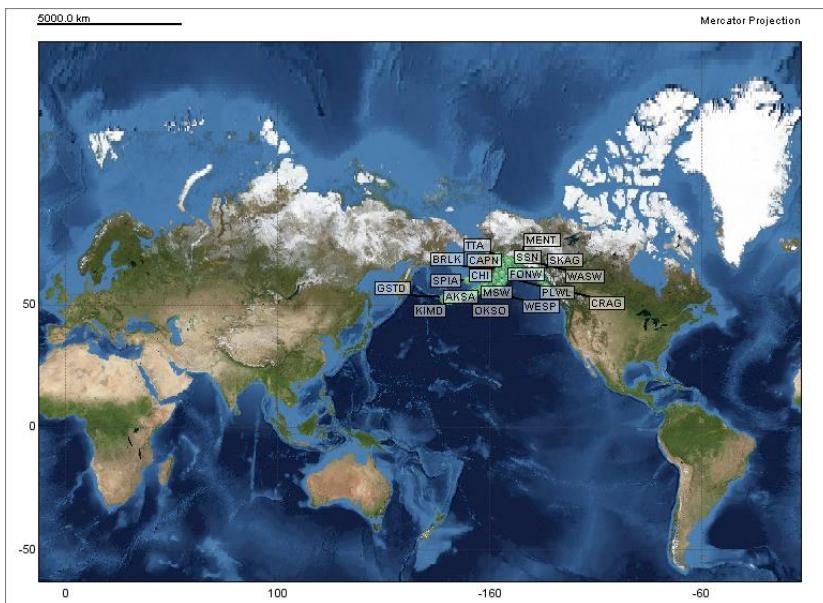
2.4.1.1 Sumber Data dan Kanal

Pada saat pertama kali dinyalakan, belum ada sumber data yang terhubung, sehingga perlu dilakukan koneksi terlebih dahulu ke server penyedia data gempa. Akan ada selektor untuk sumber data, dan selektor untuk kanal. Penjelasannya sebagai berikut. Server untuk penyedia data seismik terbagi menjadi beberapa jenis, seperti *Winston Wave Server*, *Seedlink Server*, dll. Masing-masing server terhubung dengan banyak sensor. Selektor untuk sumber data adalah selektor untuk server yang akan digunakan. Satu server biasanya terhubung dengan sensor-sensor di satu daerah tertentu. Jadi dengan kata lain, selektor sumber data memilih daerah tertentu di dunia yang hendak dijadikan fokus. Selektor kanal berguna untuk memilih satu sensor tertentu yang akan dijadikan fokus.

Saat salah satu server dipilih, maka selektor kanal akan dipenuhi dengan sensor-sensor yang tersedia. Di sebelah nama sensor akan ada indikator apakah sensor tersebut aktif atau tidak, yang ditandai dengan sebuah lingkaran kecil. Lingkaran yang berwarna hijau artinya sensor tersebut aktif. Lingkaran hijau pucat artinya sensor aktif, namun data terakhir dari sensor tersebut lebih dari 5 detik yang lalu, yang artinya ada masalah dalam mengirim beberapa data terakhir dari sensor tersebut. Lingkaran abu-abu artinya sensor tersebut tidak aktif.



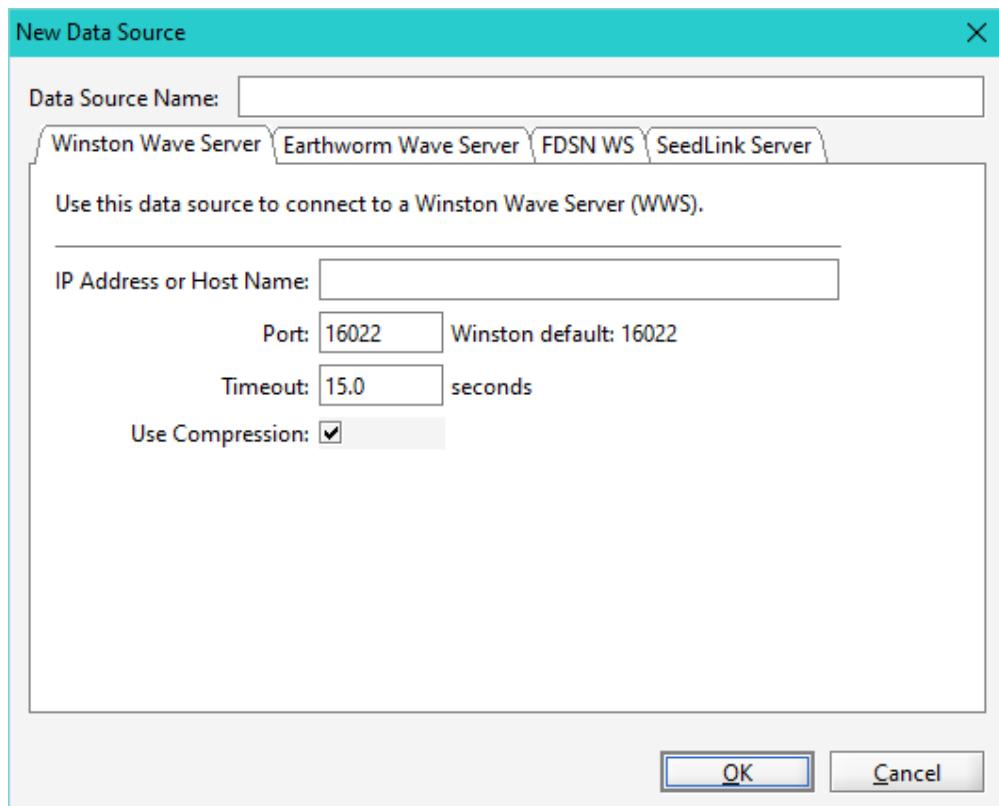
Desain untuk selektor adalah seperti gambar diatas. Pada gambar, dipilih sumber data yaitu AVO Winston, yaitu *Winston Wave Server* yang terletak di Alaska, Amerika Serikat, dengan sensor-sensor yang tersebar di sekitar daratan Alaska dan pulau-pulau Aleutian. Pada server ini, sensor-sensor yang ada dikelompokkan lagi menurut lokasi sensor-sensor tersebut berada. Agar mudah dibaca, tampilan server dan kelompoknya menurut lokasi dibuat seperti *file* dan *folder* pada komputer. Sensor juga mengirim data lokasi selain data seismik. Lokasi dari sensor akan ditampilkan di peta dunia seperti yang bisa dilihat pada gambar dibawah. Sensor akan ditandai dengan sebuah lingkaran hijau pada peta dunia. Peta dunia yang dimaksud berdasarkan proyeksi Mercator, dan disediakan secara cuma-cuma oleh USGS.



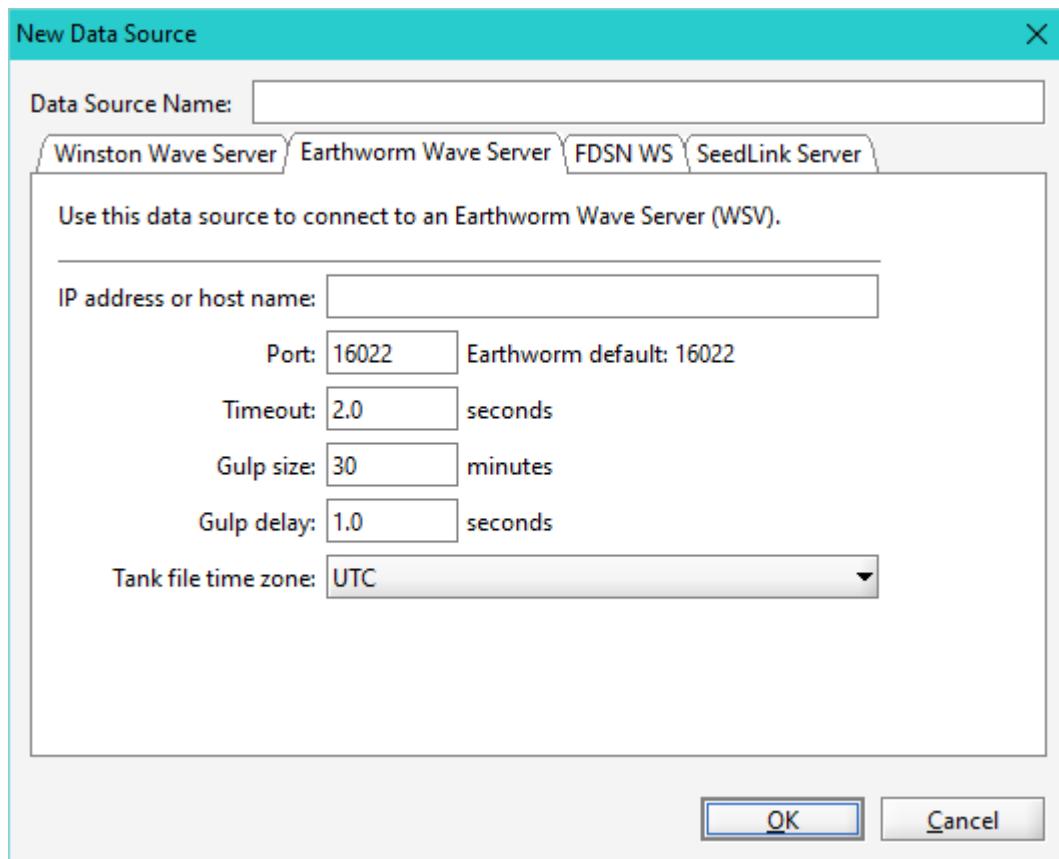
Untuk menambah sumber data baru, akan ada menu yang diwakili dengan tombol berikut



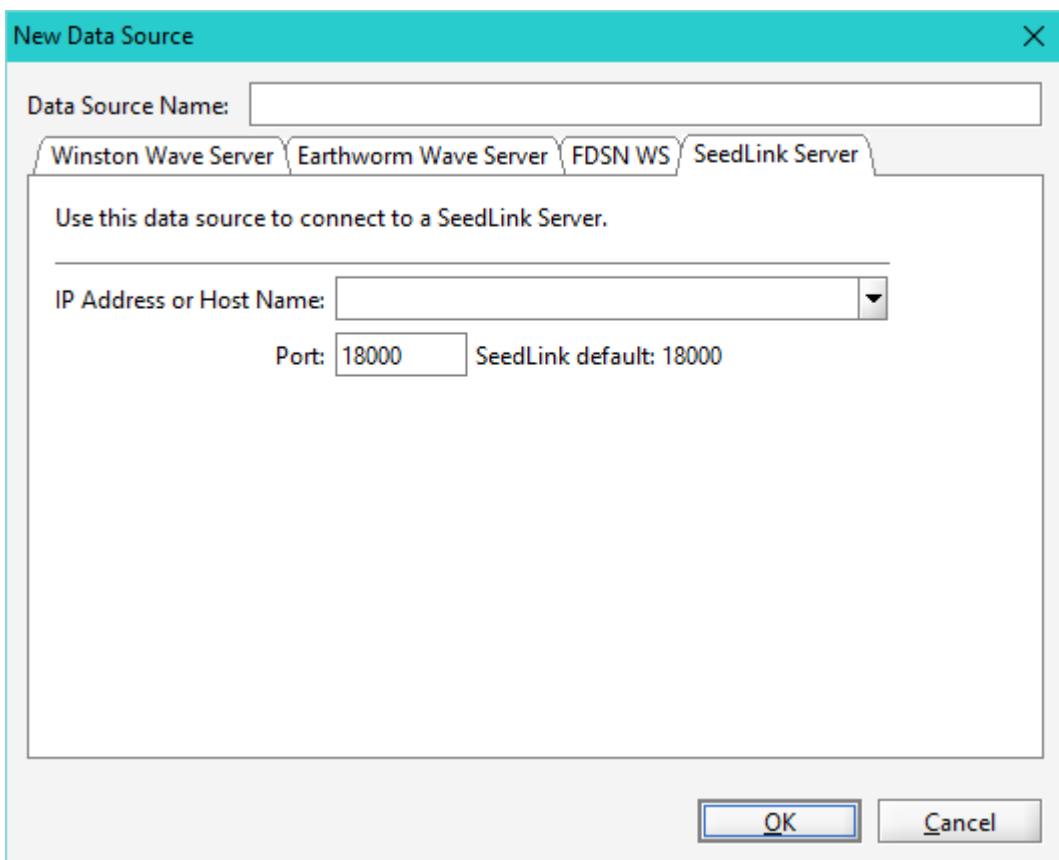
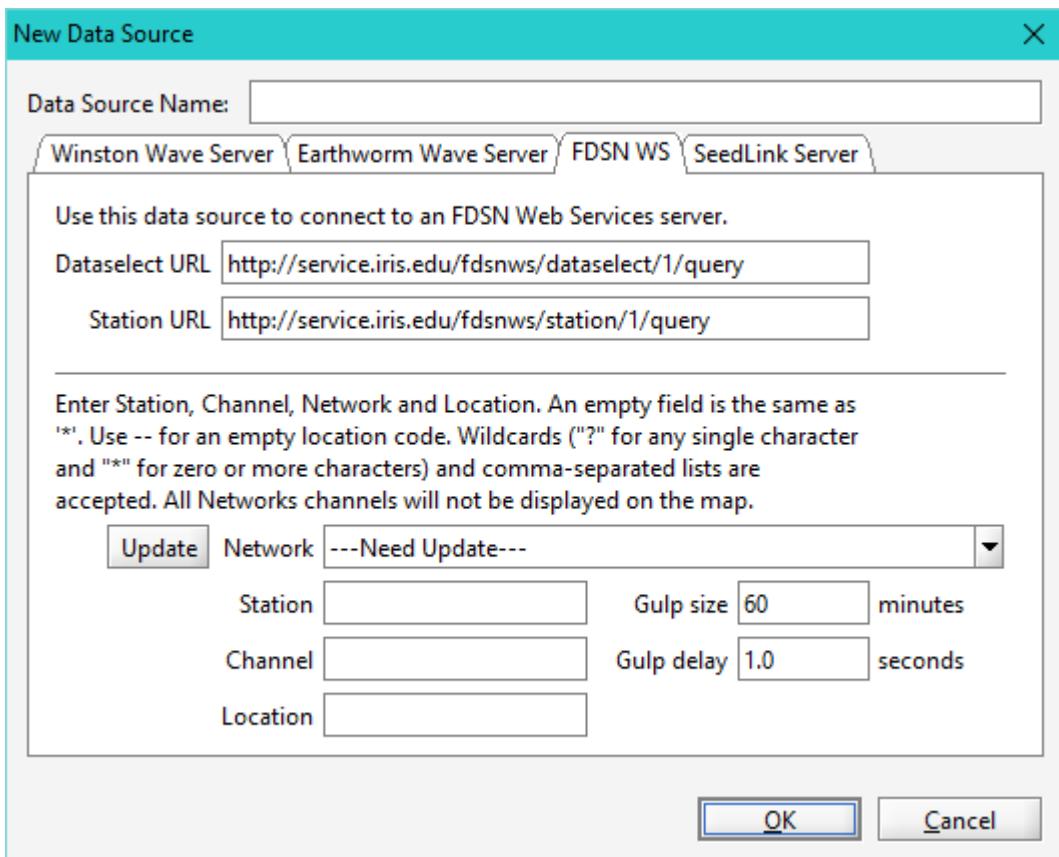
Apabila tombol tersebut ditekan, maka akan muncul kotak dialog seperti berikut. Kotak tersebut memungkinkan kita untuk menghubungkan ke server tertentu yang diketahui alamat web atau *IP Address*-nya. Ada 4 jenis yang sementara ini tersedia, yaitu Winston Wave Server, Earthworm Wave Server, FDSN WS, dan Seedlink Server. Winston adalah jenis server yang paling umum digunakan USGS. Konfigurasi ini memungkinkan program untuk langsung mendapat *stream* data dari server secara hampir *real-time*. Data yang didapat juga bisa langsung dianalisis. Untuk terhubung ke server tertentu, pengguna perlu tahu alamat IP dari server, atau *host name* dari server tersebut di internet. Tampilannya seperti gambar dibawah.



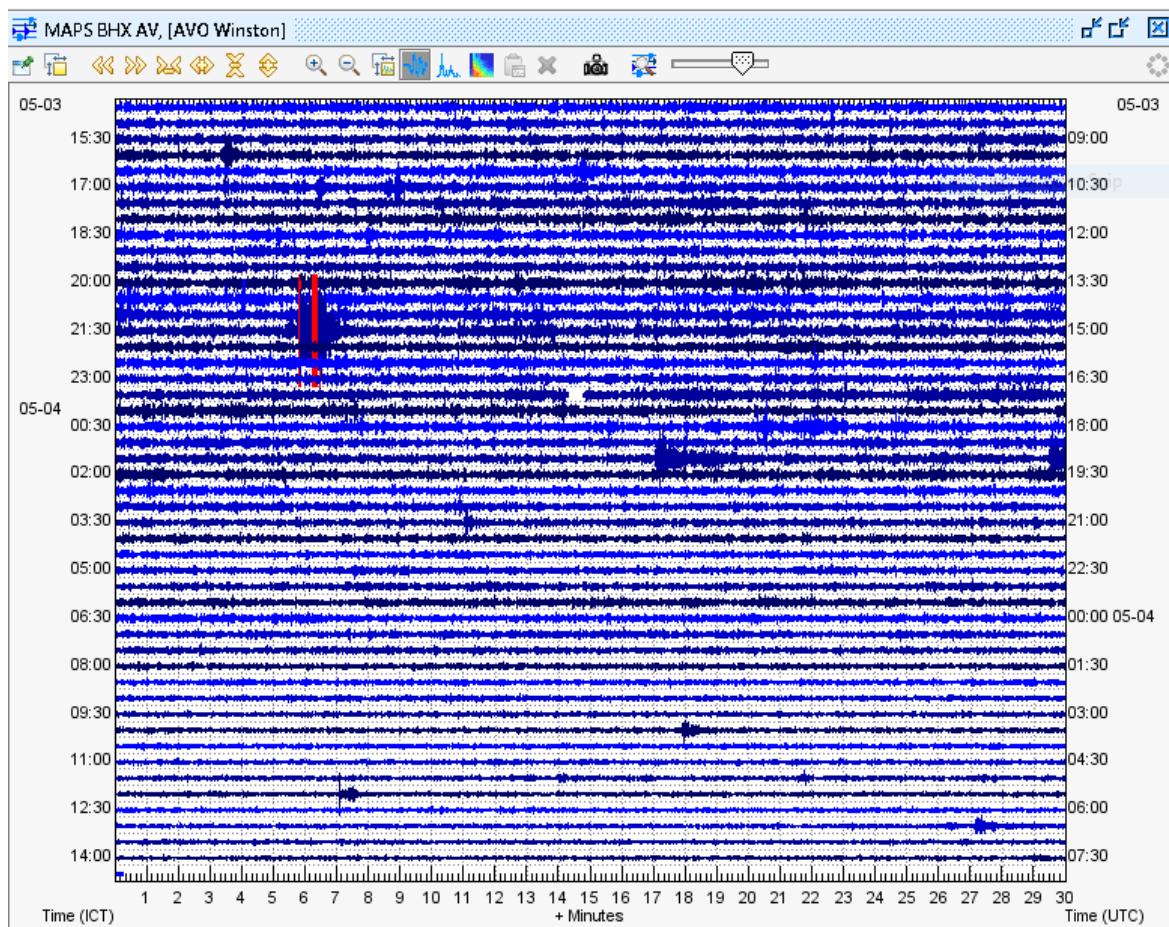
Gambar dibawah adalah pilihan untuk melakukan koneksi ke suatu *Earthworm Wave Server*. Untuk konfigurasi ini, data yang tersedia tidak bisa *real-time*, namun perlu diunduh terlebih dahulu dalam paket-paket tertentu. Setelah paket tersebut ada di komputer, maka bisa langsung dilakukan analisis. Seperti sebelumnya, perlu diketahui *IP Address* atau *host name* dari server untuk bisa melakukan koneksi. Selain itu, perlu diatur port mana di komputer yang akan melakukan koneksi. Pengaturan untuk *gulp* juga perlu dilakukan. *Gulp* adalah proses mengambil atau mengunduh data dari server *earthworm*. Pengaturan ini bertujuan untuk menentukan ukuran dan delay setiap kali melakukan *gulp* atau pengunduhan data dari internet.



Selain itu, server jenis lain yang bisa diakses adalah tipe FDSN WS dan Seedlink. FDSN (Federal Digital Seismograph Network) merupakan kumpulan jaringan seismometer yang bisa diakses melalui internet. FDSN terdiri dari jaringan yang setidaknya mengoperasikan lebih dari satu seismometer. Anggotanya berasal dari ratusan jaringan dari berbagai negara yang kemudian bekerjasama membuat format standar untuk tukar-menukar data seismik yang disebut SEED (Standard Exchange for Earthquake Data). SEED ini kemudian dikembangkan oleh Seiscomp3 menjadi Seedlink. Dengan program ini, baik sensor yang terhubung dengan jaringan FDSN maupun sensor yang beroperasi dengan protokol seedlink bisa diakses. Namun, untuk FDSN perlu diketahui alamat sensor dan alamat *data select* dari jaringan yang bersangkutan. Untuk jaringan dengan protokol seedlink, hanya perlu diketahui alamat dari server, mirip dengan konfigurasi untuk Winston Wave Server.



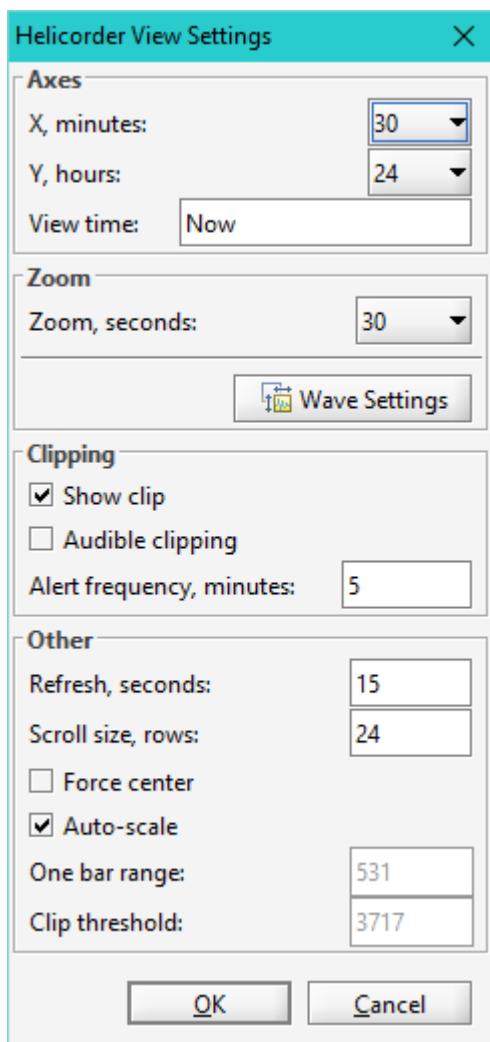
2.4.1.2 Helicorder



Helicorder adalah alat yang digunakan pada seismogram atau seismometer untuk merekam data gelombang yang didapat dari sensor. Pada zaman dahulu, helicorder mencatat gelombang dengan pensil diatas sebuah kertas. Namun, sekarang helicorder sudah diganti menjadi jenis digital, namun dengan tampilan yang dibuat mirip dengan helicorder sungguhan. Pada program ini juga dibuat tampilan data dari sensor yang dibuat mirip dengan helicorder agar memudahkan analisis oleh seismolog.

Penjelasan tampilan diatas adalah sebagai berikut. Setiap baris adalah data gelombang yang direkam selama setengah jam. Besar dari gelombang diwakili dengan warna biru, dengan warna yang semakin tua maka gelombang yang terekam pada saat itu lebih besar. Bila ada gelombang yang tidak normal terdeteksi, yaitu gelombang yang menandakan adanya gempa bumi, maka akan ditampilkan dengan warna merah. Namun, hal ini belum tentu ada gempa bumi sungguhan, karena pembacaan ini hanyalah pembacaan satu sensor. Harus ada gelombang gempa yang terdeteksi oleh banyak sensor sekaligus, sehingga gelombang tersebut dianggap sebagai gempa bumi.

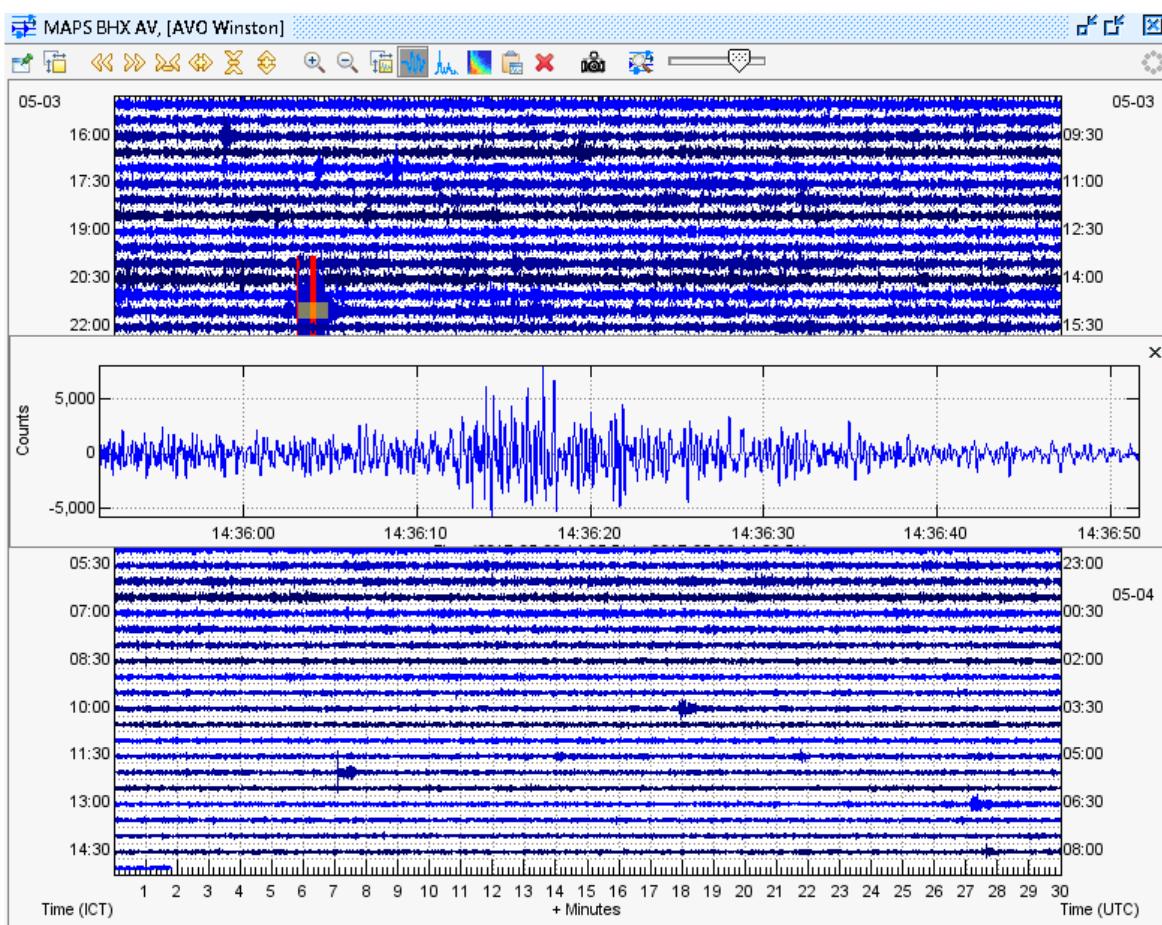
Pada sisi kiri dan kanan adalah waktu gelombang tersebut direkam. Sebelah kanan adalah waktu UTC (Coordinated Universal Time), atau waktu di garis bujur 0 derajat. Pada sebelah kiri adalah waktu lokal dari program, dalam hal ini adalah ICT (Indochina Time), atau UTC +7. Bila ada pergantian hari, juga diperlihatkan tanggal saat terjadi perpindahan hari pada zona waktu yang bersangkutan. Tampilan juga bisa diatur, contohnya untuk sumbu X, bisa dibuat agar panjang rekaman yang terlihat menjadi 1 jam atau lebih. Begitu pula dengan sumbu Y, bisa dibuat agar jumlah baris yang ditampilkan menjadi lebih banyak atau lebih sedikit. *Scaling* juga bisa dibuat otomatis atau manual. Panel lengkap untuk pengaturan tampilan helicorder dapat dilihat pada gambar berikut.



Refresh rate dari helicorder juga bisa diatur, jadi tampilan bisa dibuat seperti helicorder asli yang menampilkan data secara *real-time*. Namun, karena server USGS adalah server yang publik, ada banyak pengguna yang menggunakan data dari server tersebut. Akibatnya, merupakan suatu kebijakan USGS untuk menjaga *refresh rate* dari semua

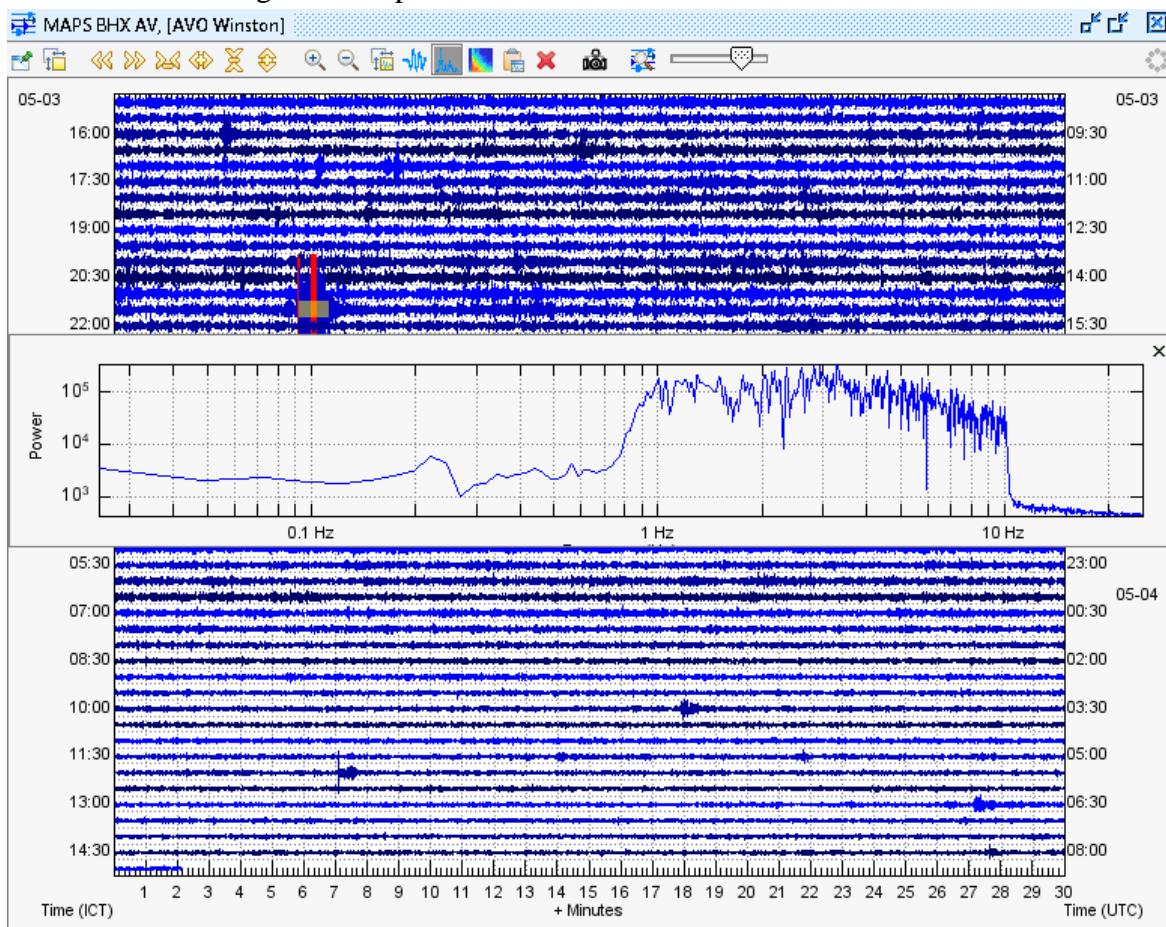
pengguna yang menggunakan datanya. Batasan itu adalah 15 detik, jadi baru tiap 15 detik sekali ada data baru yang masuk ke tampilan *helicorder*.

2.4.1.3 Tampilan Gelombang



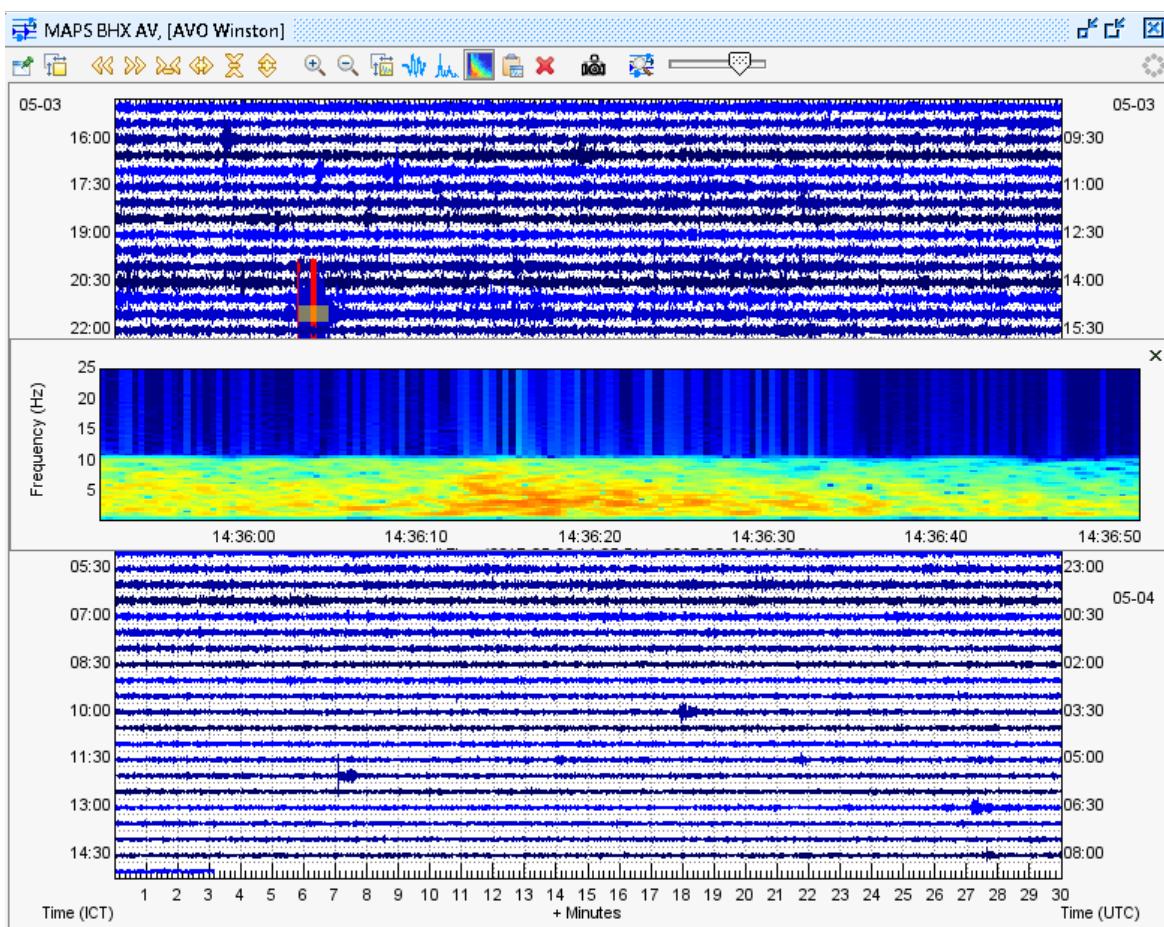
Bagian ini membahas analisis yang lebih dalam dari helicorder. Pertama, akan dibahas mengenai tampilan gelombang menurut domain waktu. Secara sederhana, fungsi ini seperti *zoom in* terhadap gelombang yang ada pada helicorder. Tampilan gelombang dibuat lebih besar, dengan skala waktu yang lebih detail, dengan sumbu X mencapai sekitar 1 menit. Sumbu Y menyatakan amplituda dari gelombang yang diukur. Perlu dicatat bahwa amplituda ini berbeda dengan amplituda gempa yang biasa dikenal, meskipun keduanya berhubungan. Pada contoh diatas, gelombang mencapai *count* lebih dari 5000, yang sudah dianggap sebagai gelombang yang diatas normal, dan dideteksi sebagai *event*.

Batas *count* antara *event* adalah skala 1000. Namun, skala ini bisa diubah sesuai dengan sensor yang ada. Batas ini membuka langkah ke analisis yang lebih lanjut, karena batas ini digunakan sebagai parameter mengukur gelombang primer dan sekunder. Nantinya, informasi ini akan berguna untuk menentukan lokasi, magnitudo, dan kedalaman gempa bumi. Sebagai contoh, pada gambar diatas, gelombang yang diambil sebagai *event* adalah yang berada diantara waktu 14:36:10 dan 14:36:20. Apabila ada *event* seperti itu dideteksi dalam sensor lain yang berdekatan, maka akan dilakukan perhitungan mengenai magnitudo dan lokasi dari gempa. Magnitudo sebenarnya bisa dihitung dengan satu sensor saja, namun informasi tersebut hanya relevan apabila ada gempa sungguhan, yang hanya bisa ditentukan dengan beberapa data dari sensor.



Gambar diatas adalah hasil analisis magnitudo dari suatu gelombang yang dipilih. Gelombang yang dipilih dapat dilihat pada bagian atas, yang diberi *highlight* berwarna kuning. Proses penentuan magnitudo adalah sebagai berikut. Dari gelombang yang didapat pada contoh sebelumnya, dilakukan transformasi Fourier untuk mendapatkan spektrum frekuensi dari gelombang tersebut. Pada gambar diatas, bagian yang di *highlight* adalah suatu *event*, sehingga *power* yang terlihat cukup besar. Tampilan frekuensi dibawah menggunakan skala logaritmik, dengan batas antara 0 sampai 20 Hz. Perlu dicatat bahwa *power* disini adalah skala yang relatif, dengan membandingkan frekuensi-frekuensi yang ada pada gelombang yang tercatat.

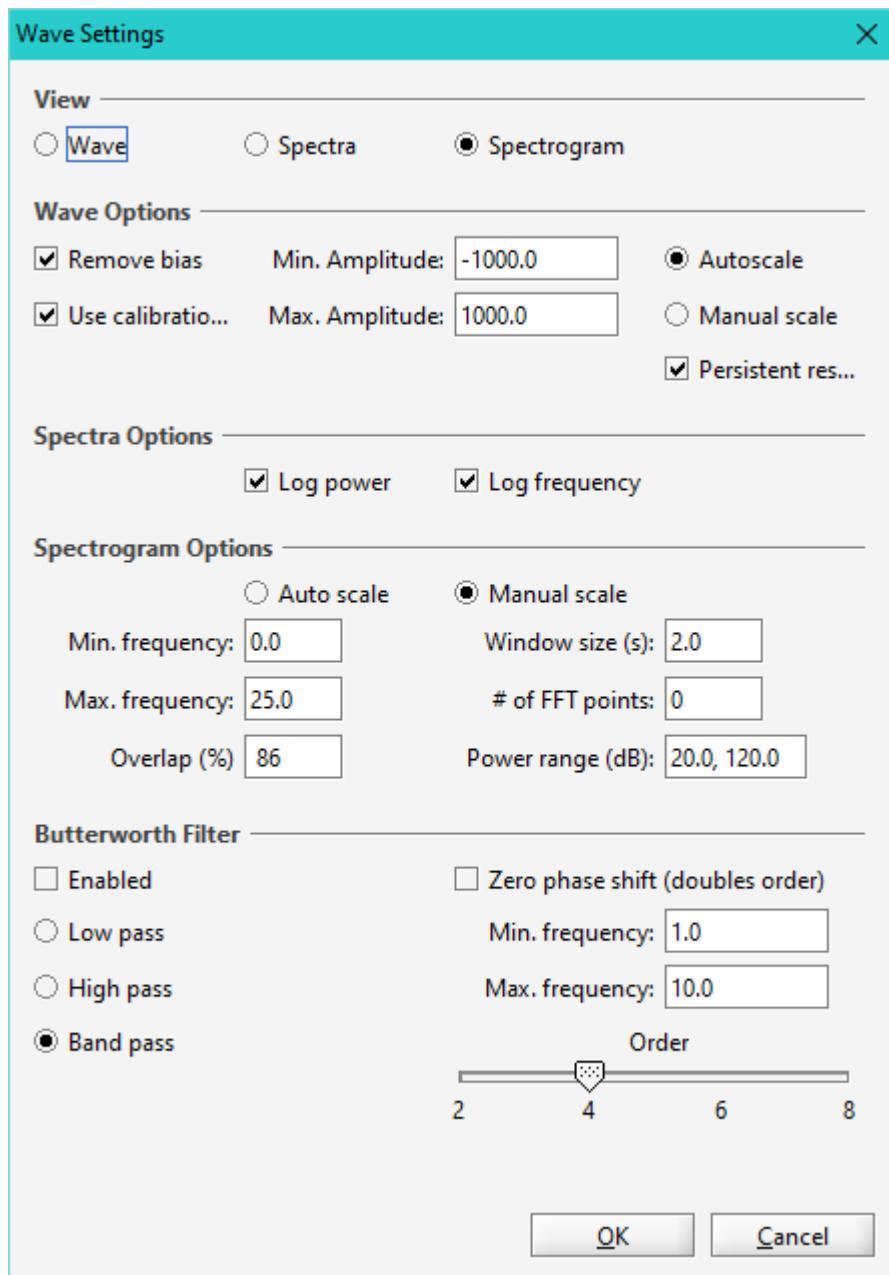
Analisis *magnitude* ini akan berguna untuk menentukan kekuatan dan lokasi dari gempa pada analisis tahap berikutnya. Perlu dicatat bahwa *magnitude* ini berbeda dengan *magnitude* gempa yang sudah dikenal. Informasi lain diperlukan untuk menentukan *magnitude* gempa, seperti lempeng tektonik tempat sensor tersebut berada, jenis tanah, dan sebagainya. Skala yang digunakan pada *magnitude* ini adalah skala relatif, artinya bila ada frekuensi dengan *magnitude* 10^4 , maka gelombang pada frekuensi tersebut mempunyai intensitas 10 kali lipat dibanding frekuensi dengan *magnitude* 10^3 . Skala ini bukan skala absolut yang mempunyai satuan, sehingga bila hanya ada satu data, belum bisa dijadikan rujukan kekuatan gempa. Selain itu, kebanyakan sensor gempa dibuat sensitif untuk frekuensi diatas 1 Hz, sehingga gelombang dengan frekuensi dibawah 1 Hz seringkali hanya *noise*.



Gambar diatas adalah hasil analisis berupa spektrum frekuensi yang digabung dengan waktu. Jadi, bisa dilihat kekuatan dari masing-masing frekuensi pada waktu tertentu. Kekuatan yang paling kecil digambarkan dengan warna biru tua, sedangkan kekuatan yang paling besar dengan warna merah tua. Semua kekuatan lain diwakili dengan warna-warna diantara merah dan biru. Tampilan ini dapat dianalogikan dengan gabungan dua tampilan sebelumnya. Dengan tampilan ini, bisa dilihat kapan terjadinya gelombang yang dianggap sebagai *event*, sekaligus dengan frekuensi dimana *event* tersebut terjadi.

Spektrum frekuensi ini juga berguna untuk analisis mengenai kekuatan gempa serta lokasi serta kedalamannya. Dengan tampilan ini, bisa diketahui *event* tertentu dengan frekuensi tertentu sampai di sensor pada waktu tertentu. Dari informasi ini bisa diketahui jarak sensor dari pusat gempa, dengan memperhitungkan sensor mana yang pertama kali mendekripsi, sensor mana yang mendekripsi berikutnya, serta perbedaan waktu antara kedua sensor tersebut. Dengan data mengenai tanah tempat sensor berada, bisa diketahui cepat rambat gempa pada tanah tersebut dengan frekuensi tertentu. Pada akhirnya, bisa ditentukan aproksimasi dari pusat gempa, dengan menghitung irisan dari lingkaran jarak masing-masing sensor.

Selain itu, karena perbedaan karakteristik gelombang P dan S, juga bisa dilihat mana gelombang P dan S dari tampilan ini. Gelombang P adalah gelombang transversal yang hanya merambat melalui permukaan bumi. Gelombang P terjadi sebelum gelombang S, sehingga bisa dilihat di tampilan ini yang mana gelombang P. Pada gelombang P, tidak terjadi perubahan frekuensi, hanya bisa dilihat waktu kedadangannya pada semua sensor. Gelombang S bisa merambat melalui lapisan bumi yang lebih dalam, sehingga dalam jarak jauh, waktu kedadangannya di semua sensor tidak linear. Informasi ini dapat digunakan untuk menentukan kedalaman gempa bumi. Gempa bumi yang lebih dalam memiliki nonlinearitas yang lebih tinggi, karena gelombang S membutuhkan waktu yang lebih lama untuk merambat melalui mantel bumi ke permukaan. Tampilan hasil analisis lanjut ini akan dilihat pada bagian 2.4.1.5.



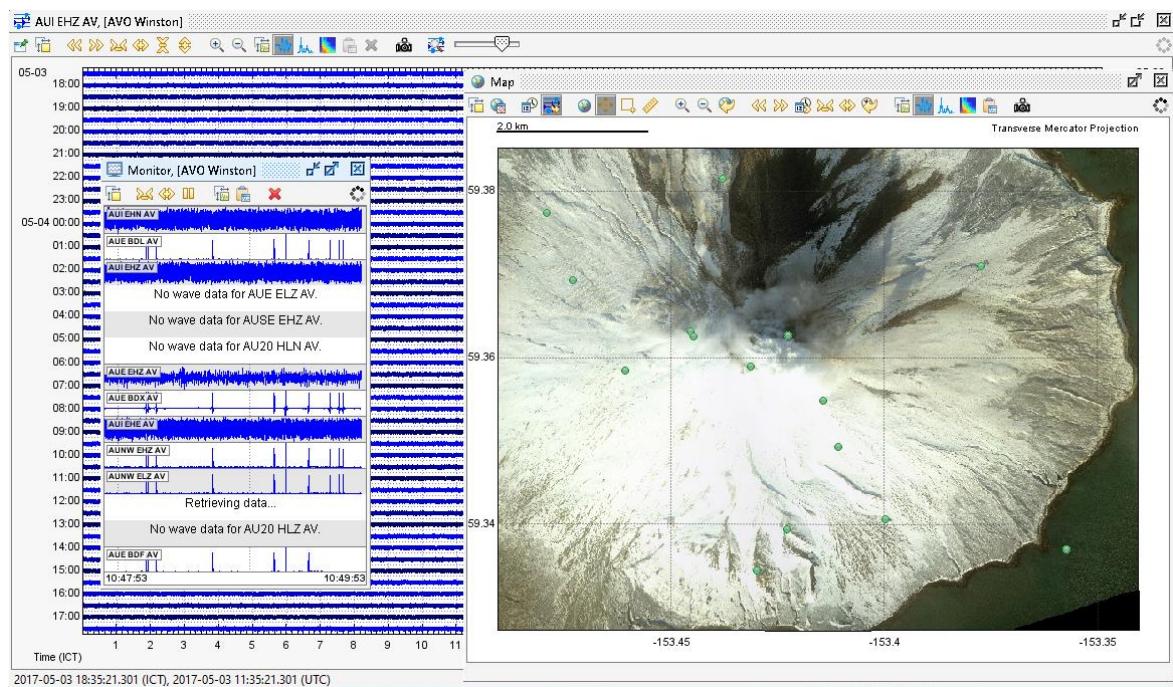
Tampilan diatas adalah pengaturan dari jendela tampilan gelombang. Sebagaimana yang telah disebutkan, ada 3 jenis tampilan yang bisa dilihat, yaitu *wave*, *spectra*, dan *spectrogram*. Pada *wave*, dapat diatur amplitude serta scaling. Pada *spectra*, dapat diatur skala tampilan, baik logaritmik maupun linear. Untuk *spectrogram*, bisa diatur beberapa fitur seperti minimum, maksimum, dan pengaturan tentang Fast Fourier Transform (FFT).

Distance to AUI EHZ AV	
●	AUL BHM8 AV (5.3 km)
●	AUL BHM9 AV (5.3 km)
●	AUL BHMA AV (5.3 km)
●	AUL BHME AV (5.3 km)
●	AUL BHN AV (5.3 km)
●	AUL BHZ AV (5.3 km)
●	AUL EHZ AV (5.3 km)
●	AUNW EHZ AV (5.5 km)
●	AUNW ELZ AV (5.5 km)
●	OPT EHZ AV (37.1 km)
●	CDD EHZ AV (46.8 km)
●	MMN EHZ AV (54.3 km)
●	FOSS BDF AV (61.5 km)
●	FOSS BDL AV (61.5 km)
●	FOSS BDX AV (61.5 km)
●	FOSS EHZ AV (61.5 km)
●	FOSS ELZ AV (61.5 km)
●	FONW BDF AV (62.3 km)
●	FONW BDL AV (62.3 km)
●	FONW EHZ AV (62.3 km)
●	FONW ELZ AV (62.3 km)
●	FOPK EHZ AV (64.2 km)
●	FOPK ELZ AV (64.2 km)
●	PDB EHZ AV (65.9 km)
●	ILS BHE AV (72.1 km)

Tampilan diatas adalah daftar sensor yang berada di jangkauan sensor yang sedang menyediakan tampilan. Sensor dengan warna hitam adalah sensor yang saat ini aktif, sedangkan dengan warna abu-abu adalah sensor yang tidak aktif. Dengan tampilan seperti ini, bisa dilihat jarak sensor, dan bila dilihat data dari sensor dengan jarak tertentu, dapat dilihat dan diperkirakan apakah *event* yang ada adalah gempa sungguhan atau tidak, berdasarkan ada atau tidaknya event yang sama di sensor yang berdekatan. Selain itu, juga bisa diperkirakan pusat gempa, dengan melihat waktu kedatangan *event* di masing-masing sensor.

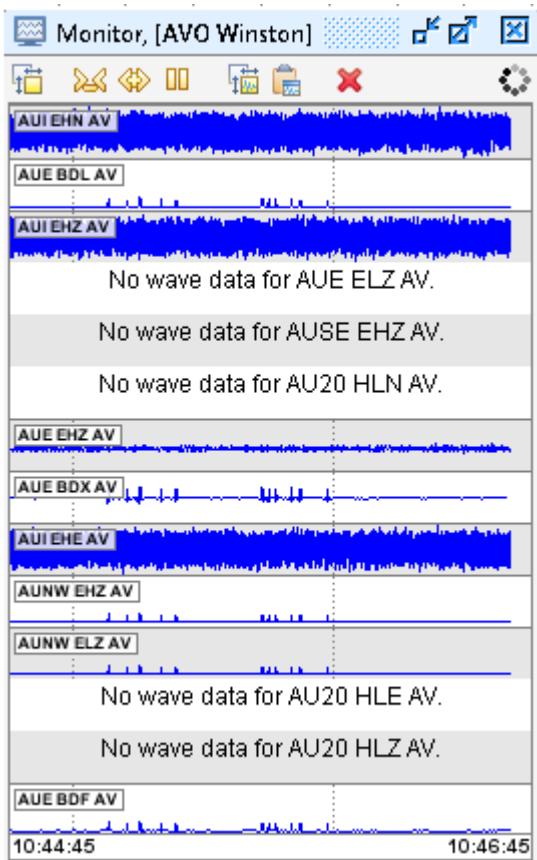
Tentu saja proses ini akan memakan waktu apabila dilakukan secara manual. Maka itu, fungsi-fungsi dari Seiscomp3 digunakan untuk menentukan kekuatan serta lokasi dari gempa. Tampilan diatas bertujuan untuk menampilkan pada pengguna tentang proses yang terjadi selama analisis, sehingga pengguna juga bisa belajar mengenai analisis gelombang seismik. Fungsi-fungsi yang digunakan akan dijelaskan dengan lebih lanjut pada bab berikutnya. Selain itu, fitur-fitur yang tidak ditampilkan di GUI juga akan dibahas dengan lebih dalam.

2.4.1.4 Tampilan Gelombang Real-Time



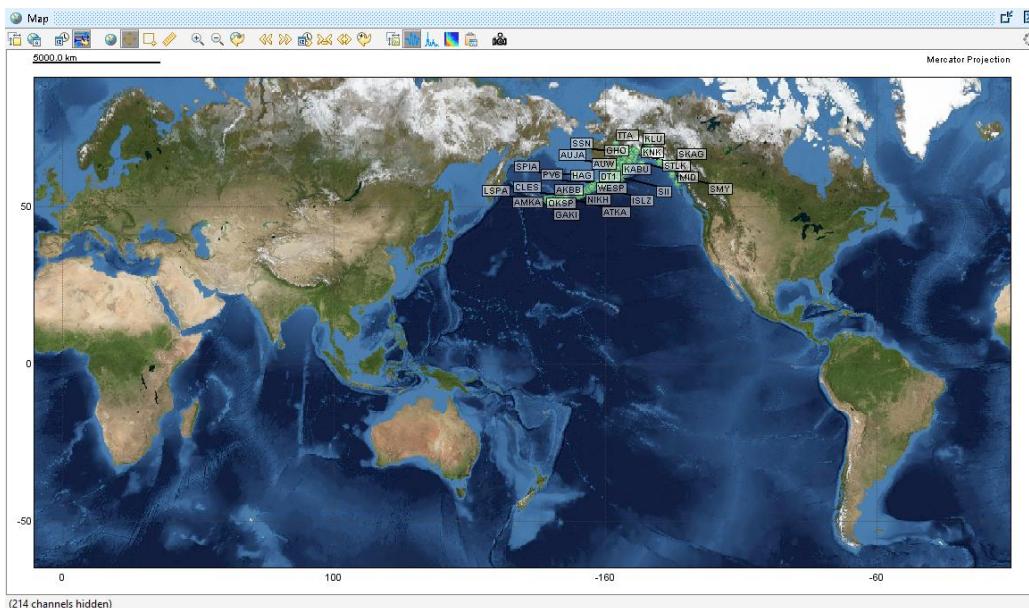
Gambar diatas adalah contoh tampilan apabila program ini hendak digunakan untuk memonitor satu lokasi kecil khusus. Pada contoh ini, suatu jaringan sensor digunakan untuk memonitor keadaan di gunung Augustine di Alaska. Tampilannya mirip dengan tampilan untuk seluruh dunia secara keseluruhan, namun ada sekelompok sensor yang terletak di dekat gunung Augustine yang bisa ditampilkan secara bersamaan. Untuk tampilan ini, juga bisa dilakukan pemantauan secara *real-time*.

Tampilan ini memungkinkan pemantauan suatu lokasi tertentu menjadi lebih mudah dan akurat. Lokasi yang membutuhkan pemantauan khusus seperti ini seperti gunung berapi dan/atau tempat yang memiliki aktivitas geologi yang tinggi. Selain gunung berapi, daerah pertambangan juga bisa menggunakan sistem pemantauan seperti ini agar kejadian lokal, seperti yang terjadi karena ledakan, bisa dengan mudah dipantau. Selain itu, sistem ini juga menggunakan algoritma khusus yang lebih sederhana untuk melakukan analisis data seismik, karena lokasi yang lebih kecil. Salah satu akibatnya adalah sulitnya membedakan gelombang P dan S karena keduanya akan datang hampir bersamaan.

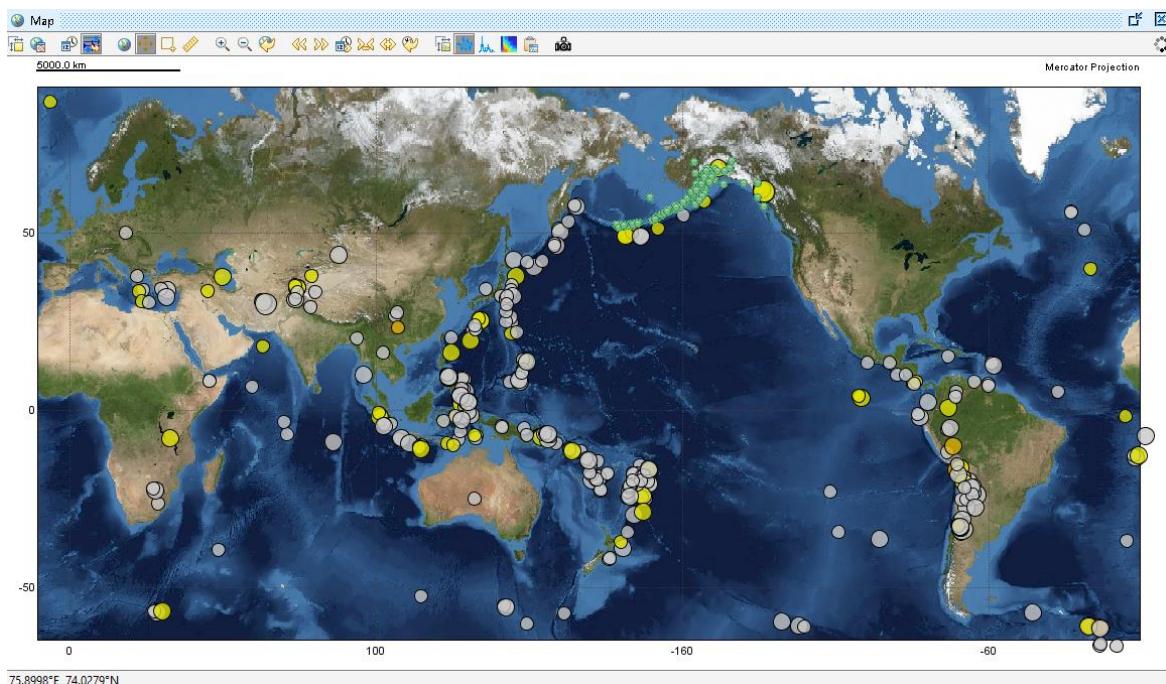


Gambar diatas adalah tampilan sensor satu kelompok dari gambar sebelumnya yang diperbesar. Ada 15 sensor yang dipasang di sekitar pulau Augustine, namun tidak semuanya aktif. Juga terdapat perbedaan skala pada beberapa sensor, seperti sensor AUI EHE AV, yang seakan-akan memiliki *event*, namun realitanya tidak ada getaran sama sekali. Hal ini karena perbedaan jenis sensor yang dipasang.

2.4.1.5 Peta dan Mode Kiosk

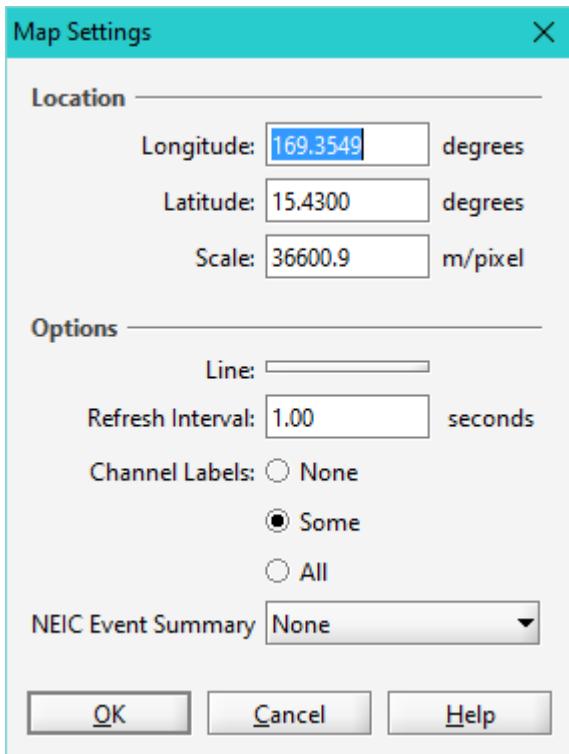


Peta pada GUI adalah peta dunia proyeksi Mercator, yang akurat dalam arti bentuk daratan dipertahankan sesuai aslinya, namun ukuran akan menyimpang di daratan yang mendekati kutub, baik utara maupun selatan. Contohnya, pada peta tampak bahwa Greenland mungkin tampak seukuran dengan Australia, padahal nyatanya Greenland jauh lebih kecil. Pada peta juga bisa ditampilkan sensor-sensor yang terhubung ke GUI, apabila terdapat pemancar GPS pada sensor tersebut. Sensor juga diberi label sesuai dengan namanya. Ada opsi untuk mematikan label, memberi label pada sebagian sensor, atau memberi label pada semua sensor.

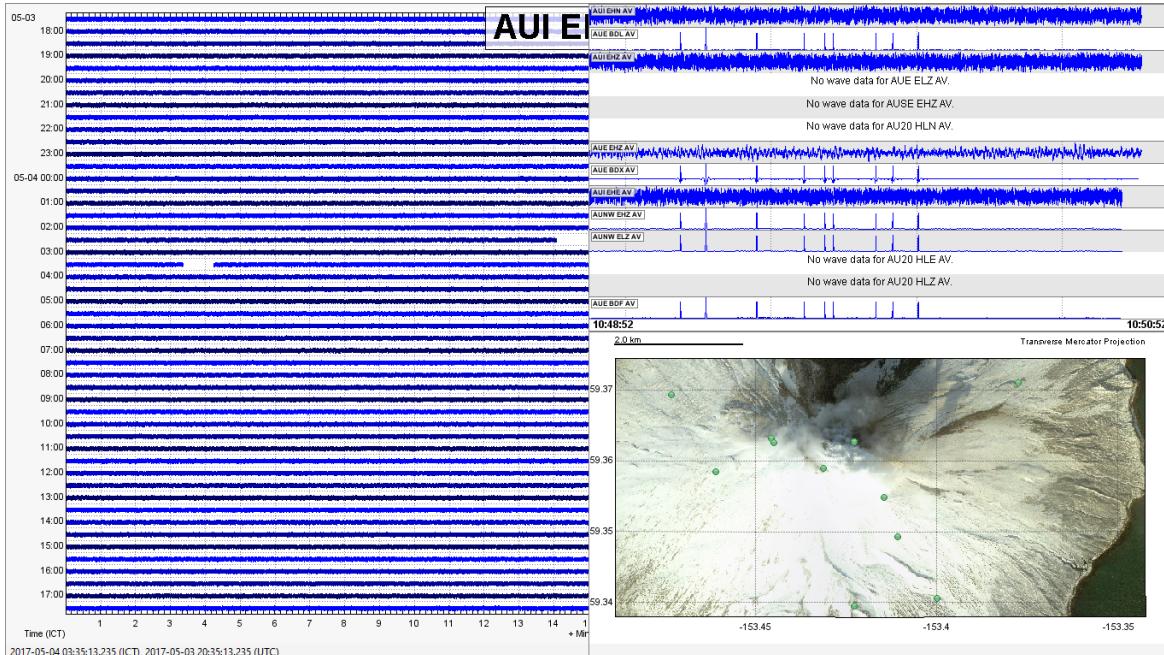


Pada gambar diatas, ditampilkan semua gempa yang ada di dunia pada bulan terakhir yang mempunyai kekuatan lebih dari 5. Untuk titik kuning maka gempa tersebut mempunyai kedalaman yang kurang dari 20 km, sedangkan titik abu-abu mempunyai kedalaman lebih dari 20 km, yang cenderung tidak bahaya. Pada titik-titik di Alaska, gempa tersebut adalah hasil analisis data yang diterima dari sensor. Ada beberapa opsi untuk tampilan gempa-gempa di dunia, yaitu dalam rentang waktu satu jam, satu hari, satu minggu, atau satu bulan terakhir. Selain itu, bisa juga dipilih gempa berdasarkan kekuatannya. Bisa dipilih gempa yang signifikan saja, gempa diatas 5, diatas 2.5, atau semua gempa.

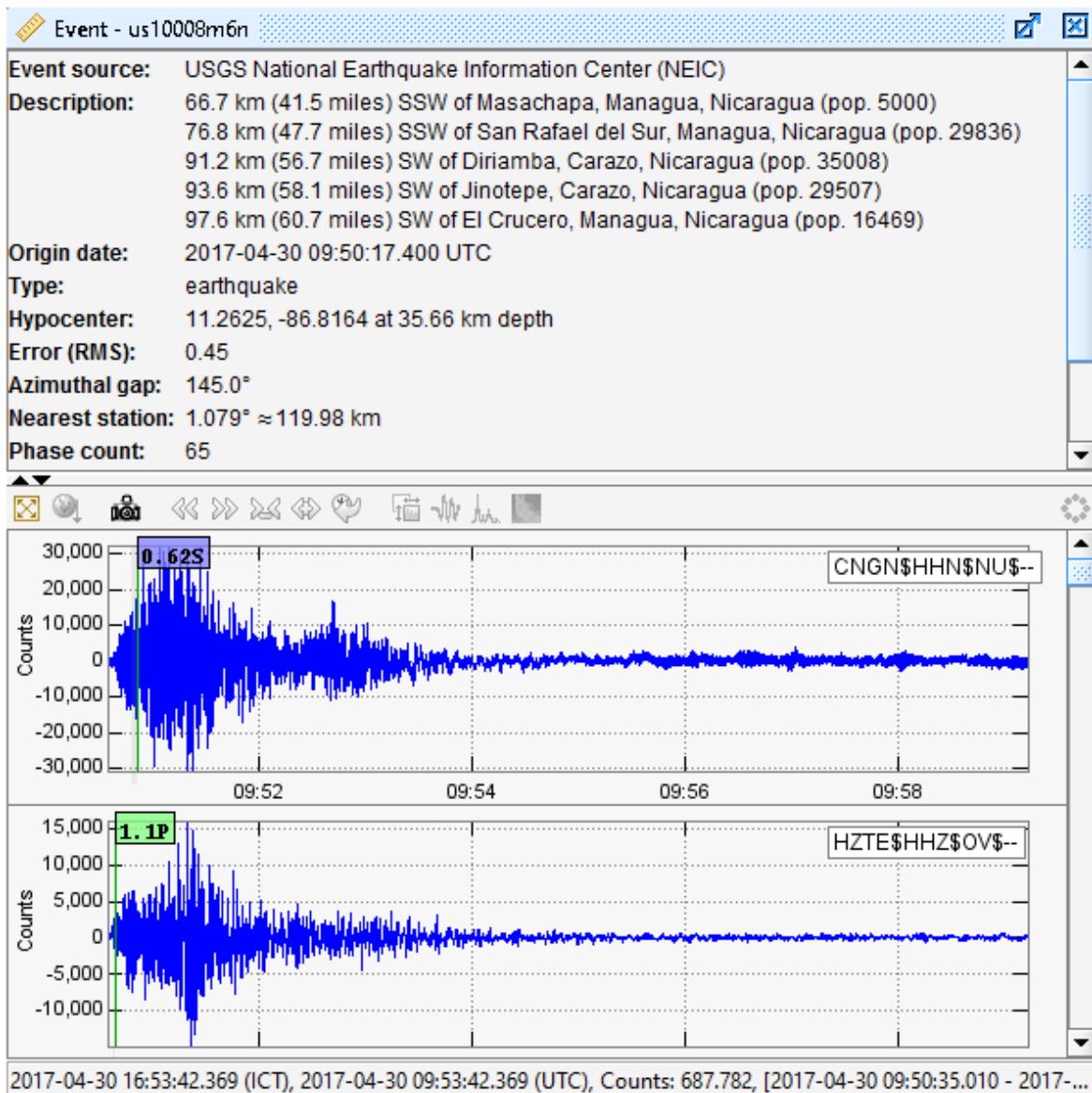
Untuk menentukan terjadinya gempa atau tidak serta menentukan kedalaman dan informasi lainnya digunakan fungsi yang disediakan oleh Seiscomp3. Pada GUI ini juga digunakan API dari USGS yang memungkinkan GUI untuk mengakses basis data gempa yang dimiliki USGS. Gabungan kedua informasi ini bisa menjadi sumber untuk pengecekan silang. Selain itu, data USGS juga berfungsi sebagai verifikasi untuk algoritma yang diterapkan pada Seiscomp3.



Gambar diatas adalah pengaturan peta. Dapat dipilih daerah yang menjadi pusat perhatian, serta *refresh rate* dari gempa-gempa yang ditampilkan di peta.



Tampilan diatas adalah mode kiosk dari GUI. Mode ini memungkinkan tampilan *full screen*. Mode ini membuat pengguna lebih mudah untuk melakukan pemantauan, terutama jika menggunakan lebih dari 1 monitor.



Tampilan diatas adalah hasil dari analisis yang dilakukan oleh fungsi Seiscomp3. Di bagian bawah adalah deteksi gelombang P dan S dari *waveform*. Pada bagian atas ditampilkan detail dari gempa yang terjadi. Sumber dari informasi gempa ini adalah sensor-sensor yang dimiliki NEIC USGS (National Earthquake Information Center), yang berada di Amerika tengah, tepatnya di dekat Nicaragua. Pada bagian berikutnya ditampilkan jarak gempa tersebut dari sensor-sensor terdekat, dengan analisis berdasarkan waktu kedatangan gempa. Selanjutnya ditampilkan pusat gempa yang dihitung berdasarkan jarak dari semua sensor, dan juga dihitung kedalamannya. Terakhir dihitung ketidakpastian dari informasi gempa seperti lokasi, yang dihitung dari faktor seperti ketidakpastian GPS, ketidakpastian tanah, serta *latency* dari server.

2.4.2 Fungsi-fungsi Seiscomp3

2.4.2.1 Scevent

Mengasosiasikan asal untuk Acara atau membentuk Acara baru jika tidak ada kecocokan yang cocok ditemukan. Memilih ukuran yang disukai.

Sebagai konsekuensi dari sistem real-time, sistem SeisComP3 menciptakan beberapa asal (hasil proses pelokalan) untuk satu gempa karena seiring berjalananya waktu lebih banyak fase seismik tersedia. Rewel menerima asal-usul ini dan mengasosiasikan asal mula kejadian. Hal ini juga memungkinkan untuk mengimpor Origins dari agensi lain. Origin Matching

Rekan-rekan yang peduli berasal dari Acara dengan mencari kecocokan terbaik dari Asal (masuk) yang baru ke Asal Lain untuk Peristiwa yang Ada. Jika pertandingan tidak ditemukan Event baru bisa terbentuk. Asal baru dicocokkan dengan Origin yang ada dengan membandingkan perbedaan lokasi (horizontal saja), beda waktu asal, dan matching Picks. Pertandingan terbaik pertama lebih disukai dimana pilihan untuk pertandingan Asal (terendah ke tertinggi).

2.4.2.2 scautoloc

Scautoloc adalah program SeisComP3 yang bertanggung jawab untuk secara otomatis menemukan kejadian seismik dalam waktu dekat. Biasanya berjalan sebagai daemon, terus menerus membaca picks dan amplitudo dan memprosesnya secara real time. Modus offline juga tersedia. Scautoloc membaca picks otomatis dan beberapa amplitudo yang terkait. Atas dasar itu, ia mencoba untuk mengidentifikasi kombinasi picks yang sesuai dengan peristiwa seismik yang umum. Jika lokasi yang dihasilkan memenuhi kriteria konsistensi tertentu, dilaporkan, kemudian diteruskan ke program lain yang mengambil asalnya sebagai masukan.

2.4.2.2.1 Location procedure

Prosedur scautoloc untuk mengidentifikasi dan menemukan kejadian seismik pada dasarnya terdiri dari langkah-langkah berikut:

- Pilih persiapan

Pada scautoloc setiap pick masuk harus disertai dengan seperangkat amplitudo tertentu. Karena pada data SeisComP3 model amplitudo dan picks adalah objek independen, amplitudo ditambahkan sebagai attribut pada picks yang sesuai saat penerimaan oleh scautoloc.

- Pilih penyaringan

Setiap pick masuk disaring, yaitu dicentang jika pick sudah usang dan jika set lengkap

amplitudo terkait sudah ada. Jika sebuah stasiun menghasilkan picks sangat sering, ini dianggap sebagai gangguan yang lebih mungkin terjadi dan menghasilkan ambang SNR yang meningkat.

- Asosiasi

Ini pertama kali mencoba mengaitkan pick masuk dengan asal-usul yang diketahui. Khusus untuk acara besar dengan lokasi yang stabil berdasarkan banyak pilihan yang sudah ada, ini adalah cara yang lebih disukai untuk menangani pick. Jika asosiasi berhasil, proses nukleasi dapat dilewati. Dalam keadaan tertentu, picks keduanya terkait dan dimasukkan ke dalam nukleator.

- Nukleasi

Jika asosiasi langsung gagal, scautoloc mencoba membuat asal baru dari ini dan pilihan lain yang tidak terkait, yang sebelumnya diterima. Proses ini disebut "nukleasi". Scautoloc melakukan pencarian grid di luar ruang dan waktu, yang merupakan prosedur yang agak mahal karena memerlukan banyak sumber daya baik dari segi CPU dan RAM. Algoritma nukleasi tambahan akan tersedia di masa depan. Grid adalah himpunan diskrit-dalam prinsip- titik-titik acak yang menggambarkan area yang diminati cukup padat. Dalam pencarian grid, masing-masing titik grid diambil sebagai hypocenter hipotetis untuk semua picks masuk. Setiap pick masuk kembali diproyeksikan tepat waktu untuk masing-masing titik grid, dengan asumsi bahwa ini adalah onset "P" pertama yang datang. Jika memilih memang sesuai dengan kedatangan event seismik "P", dan jika acara ini dicatat pada jumlah stasiun yang cukup, proyeksi baru yang diproyeksikan akan dikelompokkan dengan pick sebelumnya dari acara yang sama. Cluster akan terpadat di sekitar waktu asal di titik grid yang paling dekat dengan hypocenter. Pada prinsipnya, grid bisa jadi padat sehingga lokasi yang didapat dari pencarian grid bisa langsung digunakan. Namun, karena memori RAM dan kecepatan CPU terbatas, ini tidak mungkin dilakukan. Oleh karena itu, jika sebuah cluster diidentifikasi sebagai asal potensial, itu tidak berarti bahwa semua picks yang berkontribusi sebenarnya sesuai dengan kedatangan "P". Ini mungkin juga merupakan pertandingan kebetulan yang disebabkan oleh kekasaran grid atau kemungkinan kontaminasi oleh kebisingan yang dipetik. Oleh karena itu, program lokasi (LocSAT) dijalankan untuk mencoba lokasi dan uji coba jika rangkaian picks memang membentuk hypocenter yang konsisten. Jika memilih RMS residual terlalu besar, perbaikan dilakukan dengan mengecualikan masing-masing picks yang berkontribusi satu kali untuk menguji apakah pengurangan RMS dapat dicapai. Jika asal baru memenuhi semua persyaratan, maka diterima sebagai lokasi acara seismik baru.

Poin grid ditentukan dalam file teks "grid.txt". File default yang dikirim dengan scautoloc mendefinisikan grid dengan titik terdistribusi global di permukaan, dan titik kedalaman terbatas pada daerah yang dikenal dengan seismisitas. Ini mungkin dimodifikasi, namun tidak boleh terdiri dari terlalu banyak titik grid (> 3000 , tergantung kecepatan CPU dan RAM). Lihat di bawah untuk rincian lebih lanjut tentang file grid.

- Penyempurnaan asal

Asal usul yang dihasilkan atau diperbarui melalui hubungan dan / atau nukleasi mungkin masih terkontaminasi oleh fase yang salah ditafsirkan sebagai pendatang "P". Scautoloc mencoba memperbaiki asal-usul ini berdasarkan mis. Pilih SNR dan amplitudo. Dalam langkah pemrosesan ini, juga mencoba mengaitkan fase yang menyelinap selama upaya asosiasi pertama, mis. Karena lokasi awalnya salah Jika asal berisi jumlah kedatangan yang cukup untuk mengasumsikan hasil lokasi yang cukup baik, scautoloc juga mencoba mengasosiasikan picks sebagai fase sekunder seperti "pP". Fase sekunder semacam itu hanya "terkait dengan lemah", yaitu fase-fase ini tidak digunakan untuk lokasi. Bagi analis, bagaimanapun, berguna untuk memiliki kemungkinan "pP" fase yang telah ditentukan sebelumnya.

- Penyaringan asal

Proses ini melibatkan pemeriksaan konsistensi akhir dari asal-usul baru / yang diperbarui, dll. Selama prosedur ini, asal-usul tidak dimodifikasi lagi.

Dalam perjalanan nukleasi dan asosiasi, serta penyempurnaan dan penyaringan asal, kriteria heuristik tertentu diterapkan untuk membandingkan "kualitas" asal mula yang bersamaan. Kriteria ini digabungkan dalam nilai asal internal, yang didasarkan pada sifat pengambilan sendiri dalam konteks asal (residu, RMS, azimuthal gap) masing-masing. Selain itu, amplitudo memberikan cara yang berharga untuk membandingkan kualitas asal. Jelas, sebuah pick dengan SNR tinggi kemungkinannya akan menjadi semburan suara sementara daripada pick yang melebihi ambang SNR. Sebuah pick SNR tinggi sehingga meningkatkan nilai asal. Demikian pula, pick yang terkait dengan amplitudo absolut yang besar lebih cenderung sesuai dengan onset seismik yang sebenarnya, terutama dalam kasus observasi amplitudo besar simultan di stasiun tetangga. Sebuah kasus khusus muncul, ketika beberapa stasiun di dekatnya melaporkan amplitudo di atas ambang "XXL" tertentu. Untuk rinciannya lihat bagian "Asal mula awal". Amplitudo yang digunakan oleh scautoloc adalah tipe "snr" dan "mb", sesuai dengan amplitudo SNR (relatif, unit-less) dan amplitudo "mb" (absolut). Kedua amplitudo ini disediakan oleh scautopick. Dalam kasus setup di mana scautopick diganti dengan pemetik otomatis yang berbeda, kedua amplitudo ini harus diberikan pada scautoloc. Jika tidak, picks tidak digunakan. Pada saat ini adalah persyaratan yang ketat, kedepannya bisa berubah.

- File grid

File konfigurasi grid terdiri dari satu baris per titik grid, masing-masing titik grid yang ditentukan oleh 6 kolom, mis.:

-10.00 105.00 20.0 5.0 180.0 8

Kolom tersebut adalah koordinat titik titik (lintang, bujur, kedalaman), diameter, jarak

stasiun maksimum dan jumlah tangkapan minimum. Garis di atas menetapkan titik grid yang berpusat pada 10° S / 105° BT pada kedalaman 20 km. Hal ini sensitif terhadap kejadian dalam 5° dari pusat. Stasiun dalam jarak hingga 180° dapat digunakan untuk melakukan nukleasi suatu peristiwa. Setidaknya 8 pilihan harus berkontribusi pada asal di lokasi ini. Diameternya harus dipilih cukup besar agar sel grid bisa tumpang tindih, tapi tidak terlalu besar. Ukuran juga menentukan jendela waktu untuk mengelompokkan pick dalam pencarian grid. Jika jendela waktu terlalu lama maka risiko kontaminasi dengan picks salah meningkat. Jarak stasiun maksimum memungkinkan untuk membatasi ke stasiun tertentu untuk titik grid yang sesuai. Misalnya. Stasiun dari Australia biasanya tidak diharuskan menciptakan sebuah acara di Eropa. Jika ada keraguan, atur nilainya menjadi 180. Jumlah minimum menentukan berapa jumlah pengambilan yang diperlukan untuk titik grid tertentu agar memungkinkan pembuatan asas baru. File grid default berisi grid global dengan jarak tanam $\sim 5^{\circ}$ dengan titik tambahan pada kedalaman yang lebih tinggi di mana peristiwa fokus dalam diketahui terjadi.

- File konfigurasi stasiun

File konfigurasi stasiun berisi baris yang terdiri dari kode jaringan, kode stasiun, flag penggunaan (0 atau 1) dan jarak nukleasi maksimum. Tanda pemakaian 1 menunjukkan stasiun harus digunakan oleh scautoloc. Jika tidak digunakan, 0 harus ditentukan di sini. Jarak nukleasi maksimum adalah jarak (dalam derajat) dari stasiun sampai stasiun ini dapat berkontribusi pada asal baru. Jika jarak ini 180° , stasiun ini dapat berkontribusi pada asal-usul baru di seluruh dunia. Namun, jika jaraknya hanya 10° , jangkauan stasiun ini terbatas. Ini adalah pengaturan yang membantu dalam kasus stasiun biasa-biasa saja di wilayah di mana terdapat banyak stasiun yang bagus dan terpercaya di dekatnya. Stasiun ini kemudian tidak menimbulkan risiko terhadap lokasi yang dihasilkan di luar jarak nukleasi maksimum. Kode jaringan dan stasiun mungkin wildcard (*) untuk kenyamanan. Misalnya.:

* * 1 90

GE * 1 180

GE HLG 1 10

TE RGN 0 10

Contoh di atas berarti bahwa semua stasiun dari semua jaringan secara default dapat membuat peristiwa baru dalam 90° . Stasiun GE dapat membuat acara dari jarak jauh, kecuali stasiun HLG yang agak bising di jaringan GE, yang dibatasi hingga 10° . Dengan menetapkan kolom ke-3 menjadi 0, TE RGN diabaikan oleh scautoloc.

- Preliminary origins

Biasanya, scautoloc tidak akan melaporkan asal-usulnya dengan jumlah fase pendefinisian tertentu (ditentukan oleh autoloc.minPhaseCount), biasanya 6-8 fase. Dalam kasus peristiwa yang menghasilkan amplitudo yang sangat besar pada sejumlah stasiun yang cukup (selanjutnya disebut "peristiwa XXL"), adalah mungkin untuk menghasilkan asal mula awal yang didasarkan pada jumlah yang kurang.

Prasyaratnya adalah bahwa semua picks ini memiliki amplitudo dan SNR luar biasa yang luar biasa dan berada dalam wilayah yang relatif kecil. Picks tersebut selanjutnya disebut "XXL picks". Pick dipilih secara internal sebagai "XXL pick" jika amplitudonya melebihi ambang batas tertentu (ditentukan oleh autoloc.thresholdXXL) dan memiliki $\text{SNR} > 8$. Untuk pengambilan SNR yang lebih besar dengan amplitudo yang lebih kecil dapat mencapai tag XXL, karena dibenarkan untuk diobati. Sebuah pick SNR besar sebagai XXL pick bahkan jika amplitudonya agak di bawah ambang amplitudo XXL. Kriteria XXL harus dinilai sebagai solusi untuk mengidentifikasi picks yang membenarkan nukleasi asal mula.

- Pembalakan

Scautoloc menghasilkan dua jenis file log: file log aplikasi biasa yang berisi pemrosesan dan riwayat lokasi dan log pilihan opsional. Log pengambilan berisi semua picks yang diterima dengan amplitudo yang terkait dalam file teks sederhana, satu entri per baris. Log pilihan ini harus selalu aktif karena memungkinkan pemutaran untuk pemecahan masalah dan optimalisasi scautoloc. Jika sesuatu tidak berjalan seperti yang diharapkan, memutar ulang log pilihan akan memberi cara yang berguna untuk menemukan sumber masalahnya tanpa perlu mengolah bentuk gelombang mentah lagi. File log aplikasi berisi informasi aneka dalam format variabel. Format entri bisa berubah kapan saja, jadi tidak ada aplikasi hilir yang harus bergantung padanya. Ada beberapa baris khusus, namun. Ini berisi kata kunci tertentu yang memungkinkan penyaringan informasi penting dengan menggunakan grep. Kata kunci ini adalah NEW, UPD dan OUT, untuk yang baru, diperbarui dan keluaran asal. Mereka dapat digunakan mis. seperti

```
Grep '\(NEW \--- UPD \--- OUT \)' ~ / .seiscomp3 / log / scautoloc.log
```

Ini akan mengekstrak semua baris yang berisi kata kunci di atas, memberikan sejarah asal yang sangat sederhana (dan primitif).

2.4.2.3 scmag

Tujuan dari scmag adalah menghitung besaran. Dibutuhkan amplitudo dan asal mula sebagai masukan dan menghasilkan StationMagnitudes dan NetworkMagnencies sebagai output. Besaran yang dihasilkan dikirim ke kelompok "MAGNITUDE". Scmag tidak mengakses bentuk gelombang apapun. Ini hanya menggunakan amplitudo yang telah dihitung sebelumnya, mis. Dengan scamp. Tujuan dari scmag adalah decoupling perhitungan besarnya dari pengukuran amplitudo. Hal ini memungkinkan beberapa modul untuk menghasilkan amplitudo secara bersamaan, seperti scautopick dan scamp. Begitu asal datang, amplitudo yang terkait dengan pengambilan diambil dari buffer memori atau

database untuk menghitung besarannya. Saat ini jenis skala berikut diterapkan:

MLv

Skala lokal dihitung pada komponen vertikal dengan menggunakan istilah koreksi agar sesuai dengan standar ML

MLh

Skala lokal dihitung pada komponen horisontal untuk spesifikasi SED. Lihat mlh

Mb

Ukuran gelombang bodi band sempit menggunakan filter Butterworth urutan ketiga dengan frekuensi sudut 0,7 dan 2,0 Hz.

MB

Badan lebar pita melintang.

Mwp

Ukuran gelombang tubuh Tsuboi dkk. (1995)

Selain itu, scmag menghitung besaran turunan berikut ini:

Mw (mB)

Estimasi momen Mw berdasarkan mB menggunakan regresi Mw vs mB Bormann dan Saul (2008)

Mw (Mwp)

Estimasi momen Mw berdasarkan Mwp menggunakan Mw vs Mwp regresi Whitmore dkk. (2002).

M (ringkasan)

Ringkasan besarnya, yang terdiri dari rata-rata tertimbang dari besaran individu dan mencoba menjadi kompromi terbaik antara semua besaran. Lihat di bawah untuk konfigurasi dan juga mengingatkan bagaimana menambahkan besarnya ringkasan ke daftar besaran pilihan yang mungkin atau cara membuatnya selalu disukai.

Mw (avg)

Estimasi momen Mw berdasarkan rata-rata tertimbang besaran lainnya, saat ini MLv, mb dan Mw (mB), di masa depan mungkin besaran lainnya juga, terutama yang cocok untuk acara yang sangat besar. Tujuan Mw (avg) adalah memiliki, pada tahap apapun selama pemrosesan, perkiraan "terbaik" besarnya dengan menggabungkan semua besaran yang ada menjadi rata-rata tertimbang tunggal. Awalnya rata-rata hanya terdiri dari pengukuran MLv dan / atau mb, namun begitu pengukuran Mw (mB) tersedia, ini (dan di masa depan, magnitudo besar lainnya) menjadi semakin berat rata-rata.

2.4.2.4 scamp

Langkah-langkah scamp beberapa jenis amplitudo dari data waveform. Ini mendengarkan asal usul dan mengukur amplitudo dalam jendela waktu yang ditentukan dari titik asal. Amplitudo objek yang dihasilkan dikirim ke grup pesan "AMPLITUDE". Scamp adalah rekan scmag. Biasanya, semua amplitudo dihitung sekaligus oleh scamp dan kemudian dipublikasikan. Hanya sangat jarang amplitudo perlu dikompilasi ulang jika lokasi asal berubah secara signifikan. Amplitudo dapat digunakan kembali oleh scmag, membuat perhitungan besarnya dan update efisien. Saat ini, pemetik otomatis di SeisComP 3, scautopick, juga mengukur seperangkat kecil amplitudo (yaitu "snr" dan "mb", rasio signal-to-noise dan amplitudo yang digunakan dalam perhitungan mb magnitude, masing-masing) untuk setiap pick otomatis. Di jendela waktu tetap. Jika sudah ada amplitudo, mis. Yang sebelumnya ditentukan oleh scautopick, scamp tidak akan mengukurnya lagi untuk arus masing-masing. Amplitudo juga dibutuhkan, bagaimanapun, untuk pick manual. Scamp melakukan hal ini juga. Pilihan dengan berat lebih kecil dari 0,5 pada Origin yang sesuai dibuang.

Amplitudo untuk besaran berikut saat ini dihitung:

MLv

Skala lokal dihitung pada komponen vertikal dengan menggunakan istilah koreksi agar sesuai dengan standar ML.

MLsed

Amplitudo lokal dihitung pada horizontals. Lihat MLsed

Mb

Ukuran gelombang bodi band sempit menggunakan filter Butterworth urutan ketiga dengan frekuensi sudut 0,7 dan 2,0 Hz. Perhatikan bahwa amplitudo ini juga dihitung dengan scautopick untuk semua picks otomatis.

MB

Badan lebar pita melintang.

Mw (mB)

Estimasi momen Mw berdasarkan mB.

2.4.2.5 scqc

scqc menentukan parameter kualitas arus data seismik. Parameter output adalah parameter kontrol kualitas rata-rata (QC) dalam hal pesan kualitas gelombang. Dalam interval reguler, laporan pesan dikirim yang berisi representasi rata-rata jangka pendek dari parameter QC spesifik untuk jangka waktu tertentu. Pesan alarm dihasilkan jika rata-rata jangka pendek (misalnya 90an) dari parameter QC berbeda dari rata-rata jangka panjang (misalnya 3600) lebih tinggi dari ambang batas yang ditentukan.

Untuk menghindari beban yang berlebihan, pesan QC dikirim didistribusikan dari waktu ke waktu. Berikut parameter yang ditentukan:

Keterlambatan [s]

Perbedaan waktu antara waktu kedatangan dan catatan akhir akhir waktu ditambah setengah panjang rekaman (mean data latency, berlaku untuk semua sampel dalam catatan)

Latency [s]

Perbedaan waktu antara waktu sekarang dan catatan waktu kedatangan (feed latency)

Offset [counts]

Nilai rata-rata semua sampel dari sebuah catatan

RMS [diperhitungkan]

Nilai offset rata-rata akar offset (RMS) dikoreksi

Spike (interval [s], amplitudo [counts])

Dalam kasus terjadinya lonjakan dalam rekaman parameter ini memberikan interval waktu antara lonjakan yang berdekatan dan amplitudo rata-rata lonjakan; Secara internal daftar lonjakan disimpan (spike time, spike amplitudo); Algoritma finder spike masih pendahuluan

Gap (interval [s], panjang [s])

Dalam kasus kesenjangan data antara dua catatan berturut-turut, parameter ini memberikan interval waktu jeda dan panjang rata-rata celah

Waktu [%]

Miniseed record timing quality (0 - 100%)

2.4.2.6 scmaster

Scmaster dirancang sebagai sejenis mikrokernel atau mediator yang mendelegasikan permintaan klien. Oleh karena itu aplikasi kunci yang bertanggung jawab atas orkestrasi sistem terdistribusi. Agar dapat berpartisipasi dalam sistem terdistribusi, klien perlu mengirimkan permintaan koneksi ke master scmaster. Pada gilirannya master mengembalikan pesan pengakuan yang memberi tahu klien tentang penerimaan atau penolakannya. Jika permintaan sambung berhasil, pesan pengakuan akan memberi klien pesan kelompok yang tersedia kepadanya. Selain itu, semua klien yang terhubung saat ini akan diberi tahu tentang anggota yang baru bergabung. Jika master dikonfigurasi dengan database klien juga akan menerima pesan tindak lanjut langsung yang menyimpan alamat database ini. Alamat tersebut dapat digunakan untuk mengambil data arsip nanti. Setelah koneksi terbentuk setiap pesan akan melewati master terlebih dahulu dimana diproses sesuai dan kemudian diteruskan ke kelompok sasaran. Setelah klien selesai memproses pesan putuskan akan dikirim ke master yang pada gilirannya memberitahukan semua klien yang tersisa tentang kepergian tersebut.

Scmaster dapat dikonfigurasi dengan database untuk memastikan integritas sistem. Sebelum pesan didistribusikan oleh scmaster pesan tersebut ditulis ke database yang ditentukan. Dengan cara ini setiap pesan disimpan sebelum memasuki sistem. Jika terjadi kerusakan, semua informasi yang diperlukan dapat dipulihkan dari database. Saat ini, driver ada untuk MySQL, PostgreSQL dan sqlite. Perhatikan bahwa scmaster bisa berjalan tanpa database tapi kehilangan integritas data dalam melakukannya.

2.4.2.7 SeedLink

SeedLink? Adalah protokol akuisisi data real-time dan perangkat lunak client-server yang mengimplementasikan protokol ini. SeedLink? Protokol didasarkan pada TCP. Semua

koneksi diprakarsai oleh klien. Selama fase handshaking, klien dapat berlangganan ke stasiun dan aliran tertentu dengan menggunakan perintah sederhana dalam pengkodean ASCII. Saat handshaking selesai, aliran SeedLink? "Paket" yang terdiri dari SeedLink 8 byte? Header (berisi nomor urut) diikuti dengan rekaman Mini-SEED 512 byte, dikirim ke klien. Paket masing-masing stasiun selalu ditransfer dalam urutan tepat waktu (FIFO). SeedLink? Implementasi yang digunakan di SeisComP adalah yang tertua dan paling banyak digunakan, namun, ada implementasi lain. Implementasi terkenal lainnya dikerahkan di IRIS DMC dan beberapa produsen telah menerapkan SeedLink? Di firmware digitizer mereka Semua implementasi umumnya kompatibel, namun tidak semuanya mendukung SeedLink penuh protokol. Di sisi lain IRIS DMC mengimplementasikan beberapa ekstensi yang tidak didukung oleh server lain. Berikut ini kami menggunakan "SeedLink" Untuk menunjukkan SeedLink. Implementasi yang digunakan di SeisComP. Sumber data SeedLink Server bisa apa saja yang didukung oleh SeedLink? Plug-in - sebuah program kecil yang mengirimkan data ke SeedLink. Server Plug-in dikendalikan oleh SeedLink Server, mis., Plug-in secara otomatis dinyalakan ulang jika macet atau terjadi timeout. Data yang disertakan oleh plug-in dapat berupa paket Mini-SEED atau hanya sampel bilangan bulat mentah dengan informasi waktu yang menyertainya. Dalam kasus terakhir, SeedLink? Server menggunakan "Stream Processor" terintegrasi untuk membuat aliran data yang diinginkan dan merakit paket Mini-SEED.

2.4.2.7.1 Handshaking

Ketika koneksi TCP / IP telah ditetapkan, server akan menunggu klien memulai handshaking tanpa mengirimkan data ke klien terlebih dahulu. Selama handshaking klien mengirim SeedLink? Perintah ke server Perintah digunakan untuk mengatur koneksi ke mode tertentu, selector aliran setup, meminta nomor urut paket untuk memulai dan akhirnya memulai transmisi data. SeedLink? Perintah terdiri dari senar ASCII diikuti oleh nol atau beberapa argumen yang dipisahkan oleh spasi dan diakhiri dengan carriage return (<cr>, kode ASCII 13) diikuti oleh optional linefeed (<lf>, kode ASCII 10). Perintah dapat dibagi menjadi dua kategori: "perintah tindakan" dan "perintah pengubah". Perintah aksi melakukan fungsi seperti memulai transfer data. Perintah modifier digunakan untuk mengambil spesialisasi atau memodifikasi fungsi yang dilakukan oleh perintah aksi yang diikuti. Ketika sebuah server menerima perintah pengubah, ia merespons dengan string ASCII "OK" diikuti oleh carriage return dan feed line untuk mengetahui bahwa perintah tersebut telah diterima. Jika perintah tidak dikenali oleh server atau memiliki parameter yang tidak valid, maka string ASCII "ERROR" dikirim sebagai tanggapan terhadap klien diikuti oleh carriage return dan feed line. Klien tidak boleh mengirim perintah lebih lanjut sebelum menerima respons terhadap perintah pengubah sebelumnya. Jika terjadi kesalahan jaringan atau timeout, klien harus menutup koneksi dan memulai sesi baru ..

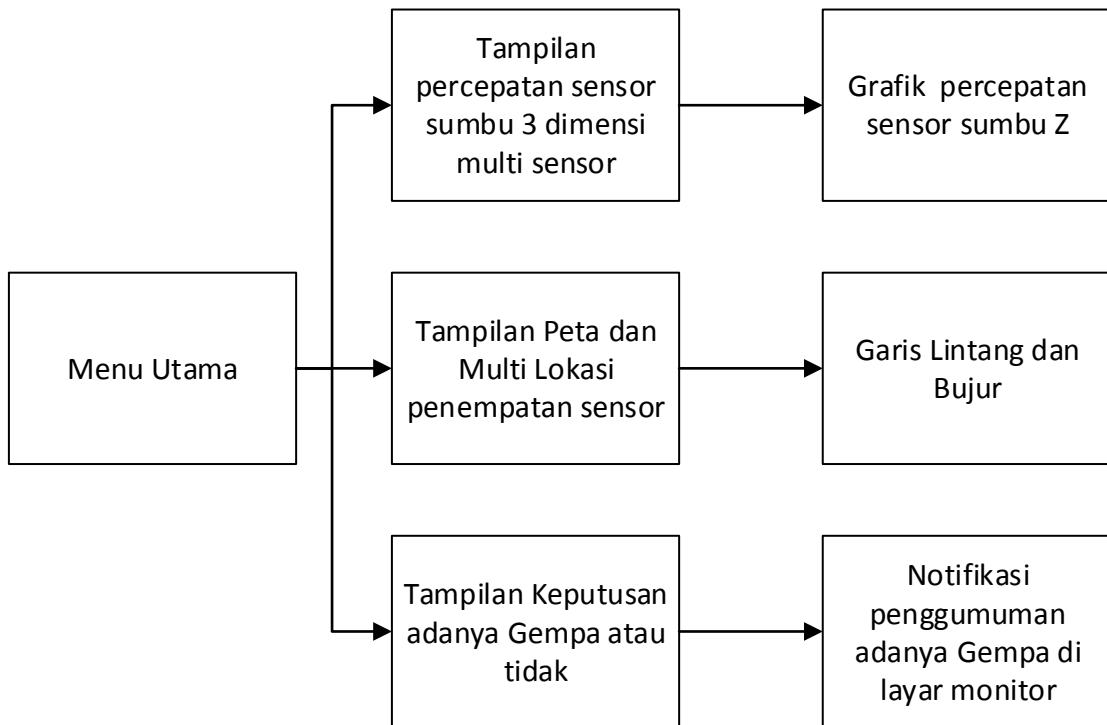
2.4.3 Graphics User Interface untuk pengguna umum

GUI yang akan dibuat berfungsi untuk memberitahu pengguna akan adanya potensi gempa atau tidak dari pengolahan data secara cepat ketika terhubung dengan jaringan ECN. Pembuatan GUI memerlukan *development tools* yang sesuai dengan kebutuhan perancangan. Pada saat ini terdapat berbagai macam *development tools* yang dapat digunakan dalam pengembangan GUI berbasis sistem operasi Microsoft Windows, namun harus dibatasi oleh Bahasa yang digunakan sesuai dengan kemampuan pengembang. Saat ini pengembang mampu menggunakan bahasa C# dalam pembuatan GUI. Sebagai contoh, program pembuatan GUI dengan bahasa C# adalah seperti Microsoft Visual Studio, dan Xamarin Studio. Selain dengan bahasa C#, ada perangkat lunak NetBeans dengan bahasa pemrograman Java yang dapat digunakan dalam pembuatan GUI. Berikut ini tabel perbandingan ketiga perangkat lunak tersebut untuk mempertimbangkan dengan perangkat lunak apa yang akan digunakan untuk melakukan pembuatan GUI khusus untuk pengguna umum.

Microsoft Visual Studio	Xamarin Studio	Netbeans
(-) Hanya dapat digunakan pada OS Microsoft Windows	(+) Dapat digunakan di Microsoft Windows, Linux, dan Mac OS	(+) Dapat digunakan di Microsoft Windows, Linux, dan Mac OS
(+) Ada versi yang tidak berbayar (Community Edition)	(-) Berbayar	(+) Tidak berbayar
(+) Menggunakan bahasa C#	(+) Menggunakan bahasa C#	(-) Menggunakan bahasa Java

Setelah dilakukan perbandingan dan pertimbangan antara kelebihan dan kekurangan masing-masing perangkat lunak untuk pembuatan GUI, maka diputuskan untuk menggunakan Microsoft Visual Studio. Hal ini disebabkan oleh penggunaan bahasa C# yang lebih familiar digunakan oleh pengembang. Apabila menggunakan Netbeans, maka pengembang harus mengerti penggunaan bahasa Java dan hal ini memakan waktu proses pengerjaan.

Kemudian setelah melakukan penentuan perangkat lunak pengembangan GUI, maka selanjutnya akan dibahas tentang fungsi yang akan ditampilkan oleh GUI. Fungsi yang akan dibuat adalah sebagai berikut:



Gambar Tampilan-tampilan yang akan dibuat pada GUI untuk pengguna umum

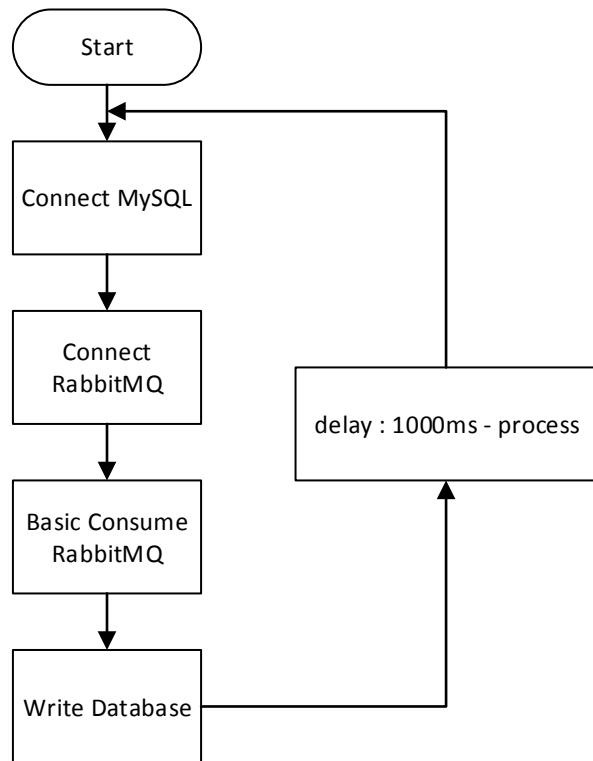
2.5 Desain Basis Data

Pada server terdapat media penyimpanan data berupa database MySQL. Database MySQL dibuat untuk menyimpan data GeoJSON atau parameter-parameter yang dibutuhkan untuk analisa gempa. Penggunaan MySQL umum digunakan dalam mengimplementasikan database sehingga desain server pada proyek ini menggunakan MySQL. Oleh karena itu, akan dibuat kolom-kolom yang sesuai dengan data GeoJSON.

Data-data tersebut meliputi:

- a. Nama sensor + *Timestamp*
- b. *Time Zone*
- c. Interval waktu pengiriman
- d. Garis Lintang
- e. Garis Bujur
- f. Percepatan sumbu x
- g. Percepatan sumbu y
- h. Percepatan sumbu z

Nama sensor beserta *timestamp* akan menjadi data yang unik sehingga bisa dijadikan *primary key* untuk pengisian *query* pada *database MySQL*. Desain program untuk dapat melakukan penyimpanan pesan menuju Database adalah sebagai berikut:



Gambar Flowchart dari program konversi pesan dari RabbitMQ menuju MySQL



INSTITUT TEKNOLOGI BANDUNG

PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 **(022)2508135-36**, **(022)2500940**
BANDUNG 40132

Dokumentasi Produk Tugas Akhir Lembar Sampul Dokumen

Judul Dokumen

TUGAS AKHIR TEKNIK ELEKTRO:
Sistem Detektor Gempa dan Peringatan Dini
Tsunami Decision Support System

Jenis Dokumen

IMPLEMENTASI

Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB

Nomor Dokumen

B400-01-TA1617.01. 069

Nomor Revisi

02

Nama File

B400 .docx

Tanggal Penerbitan

12 Mei 2017

Unit Penerbit

Prodi Teknik Elektro - ITB

Jumlah Halaman

101

(termasuk lembar sampul ini)

Data Pemeriksaan dan Persetujuan

Ditulis oleh	Nama	Kevin Shidqi	Jabatan	Anggota
	Tanggal	12 May 2017	Tanda Tangan	
	Nama	Bramantio Yuwono	Jabatan	Anggota
	Tanggal	12 May 2017	Tanda Tangan	
	Nama	Christoporus Deo	Jabatan	Anggota
	Tanggal	Putratama		
		12 May 2017	Tanda Tangan	
Diperiksa oleh	Nama	Dr. techn. Ary Setijadi	Jabatan	Dosen Pembimbing
	Tanggal	Prihatmanto		
		12 Mei 2017	Tanda Tangan	
Disetujui Oleh	Nama	Dr. techn. Ary Setijadi	Jabatan	Dosen Pembimbing
	Tanggal	Prihatmanto		
		12 Mei 2017	Tanda Tangan	

DAFTAR ISI

DAFTAR ISI.....	2
CATATAN SEJARAH PERBAIKAN DOKUMEN	3
IMPLEMENTASI PROYEK SISTEM JARINGAN DETEKTOR GEMPA DAN TSUNAMI DECISION SUPPORT SYSTEM	4
1 PENGANTAR	4
1.1 RINGKASAN ISI DOKUMEN	4
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN.....	4
1.3 REFERENSI.....	4
1.4 DAFTAR SINGKATAN	4
2 IMPLEMENTASI PROYEK.....	6
2.1 IMPLEMENTASI MESSAGING SERVER DAN DATABASE	6
2.1.1 <i>Pendahuluan</i>	6
2.1.2 <i>Struktur Implementasi</i>	6
2.1.3 <i>Environment</i>	6
2.1.4 <i>Implementasi Messaging Server RabbitMQ</i>	7
2.1.5 <i>Implementasi Database MySQL</i>	13
2.1.6 <i>Implementasi pengambilan pesan oleh Consumer</i>	15
2.2 SENSOR EARTHQUAKE CATCHER NETWORK	27
2.3 PERANGKAT LUNAK PENGOLAH DATA DAN ANTARMUKA GRAFIS	44
2.3.1 <i>Antarmuka Pengguna Grafis</i>	44
2.3.2 <i>Pengolahan Data</i>	51

Catatan Sejarah Perbaikan Dokumen

REVISI	TANGGAL	OLEH	PERBAIKAN

Implementasi Proyek Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System

1 Pengantar

1.1 Ringkasan Isi Dokumen

Dokumen B400 ini merupakan dokumentasi hasil implementasi proyek pengembangan Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System. Dokumen B400 ini terbagi dalam dua bab. Bab pertama adalah pengantar yang di dalamnya dibahas ringkasan isi dokumen, tujuan penulisan, kegunaan dokumen, referensi yang digunakan, dan daftar singkatan. Selanjutnya pada bab kedua dibahas hasil implementasi proyek pengembangan Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System. Pembahasan hasil implementasi produk tersebut meliputi implementasi pada sensor gempa, messaging server, Seiscomp3, dan GUI berdasarkan desain sistem yang telah dipaparkan pada dokumen B100, B200, dan B300

1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen

Tujuan dari penulisan dokumen B400 ini adalah:

- a. Pemaparan hasil implementasi tugas akhir Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System.;
- b. Dokumentasi dalam pelaksanaan tugas akhir Sistem Jaringan Detektor Gempa dan Tsunami Decision Support System..

Dokumen B400 ini ditujukan untuk tim dosen pembimbing TA1617.01.069 dan tim tugas akhir Teknik Elektro ITB 2016.

1.3 Referensi

- [1] Rudloff, Alexander, *German-Indonesian Tsunami Early Warning System (GITEWS) Decision Support System (DSS)*, Deutschen Zentrums für Luft- und Raumfahrt (DLR): Köln (2010)

1.4 Daftar Singkatan

SINGKATAN	ARTI

SINGKATAN	ARTI

2 Implementasi Proyek

2.1 Implementasi Messaging Server dan Database

2.1.1 Pendahuluan

Pengiriman pesan dilakukan dari sensor gempa menuju messaging server. Protokol yang digunakan oleh sensor untuk mengirim pesan adalah MQTT, sedangkan pesan diterima oleh messaging server RabbitMQ dengan protocol AMQP. Hal ini dapat dilakukan dengan menggunakan format pengiriman pesan berupa file dengan ekstensi .json, sehingga kedua protocol tersebut dapat saling berkomunikasi satu sama lain. Kemudian pesan yang sudah tersimpan dalam RabbitMQ akan di consume menggunakan console C# untuk kemudian dimasukkan ke dalam database MySQL.

2.1.2 Struktur Implementasi

Implementasi pengiriman pesan menuju messaging server RabbitMQ dilakukan berdasarkan perancangan produk yang sudah dilakukan. Sebuah sensor gempa akan mengirimkan data dengan format .json untuk dimasukkan ke dalam sebuah queue pada RabbitMQ. Namun untuk dapat memasukkan pesan dari sensor yang identik ke dalam sebuah queue, maka perlu dilakukan beberapa tahap implementasi, yaitu adalah sebagai berikut:

- a. menentukan protokol pengiriman pesan yang dilakukan oleh sensor gempa;
- b. pembuatan akun server RabbitMQ umum (CloudAMQP) di internet atau akun server RabbitMQ local yang telah disediakan;
- c. pengaturan server dengan melakukan konfigurasi queue, topic dan exchange melalui browser desktop sehingga pesan yang diterima mempunyai channel tersendiri.

Kemudian setelah itu, untuk dapat memasukkan data dari RabbitMQ menuju MySQL, maka dilakukan beberapa tahap implementasi, yaitu:

- a. membuat program yang dapat melakukan consume pada server RabbitMQ yang berkesesuaian (dalam bahasa C#, maupun Python);
- b. pengaturan kecepatan consume di dalam program hingga 1 pesan / detik supaya memiliki time stamp yang teratur;
- c. pengiriman data dari program menuju MySQL untuk disimpan di dalam database.

2.1.3 Environment

Perangkat keras dan piranti lunak yang dibutuhkan dalam mengimplementasikan RabbitMQ server adalah sebagai berikut:

- a. perangkat keras Laptop Asus X450JE sebagai komputer untuk menjalankan lokal RabbitMQ dan juga sebagai tempat Microsoft Visual Studio 2013 dan MySQL dijalankan.
- b. perangkat lunak Microsoft Visual Studio 2013 sebagai perangkat lunak IDE untuk mengembangkan GUI dan program yang berfungsi sebagai consumer RabbitMQ.
- c. perangkat keras Raspberry Pi 3 sebagai mini komputer untuk menjalankan lokal RabbitMQ

- d. perangkat lunak Python 3.5 sebagai perangkat lunak IDE yang digunakan pada Raspberry Pi 3 yang berfungsi sebagai pengirim pesan maupun consumer pada RabbitMQ
- e. perangkat keras Node MCU Amica sebagai mikrokontroller sensor gempa untuk mengirimkan pesan geospasial menuju server RabbitMQ
- f. perangkat lunak Arduino IDE 1.8.1 sebagai perangkat lunak IDE untuk mengembangkan algoritma pengiriman pesan geospasial menuju RabbitMQ
- g. JSON sebagai library dan syntax pengiriman pesan antara sensor gempa dengan RabbitMQ dan juga antara RabbitMQ dengan consumer

2.1.4 Implementasi Messaging Server RabbitMQ

Penggunaan RabbitMQ pada sebuah devais bisa dilakukan dengan melakukan instalasi dan konfigurasi pada platform tertentu. Dalam proyek ini, platform yang digunakan adalah dengan Windows, Linux, maupun Arduino. Selain menggunakan platform, dapat juga menggunakan RabbitMQ umum yang terdapat pada internet dengan menggunakan jasa CloudAMQP. Rincian implementasi RabbitMQ pada semua devais yang digunakan adalah sebagai berikut:

2.1.4.1 Implementasi RabbitMQ dengan menggunakan CloudAMQP

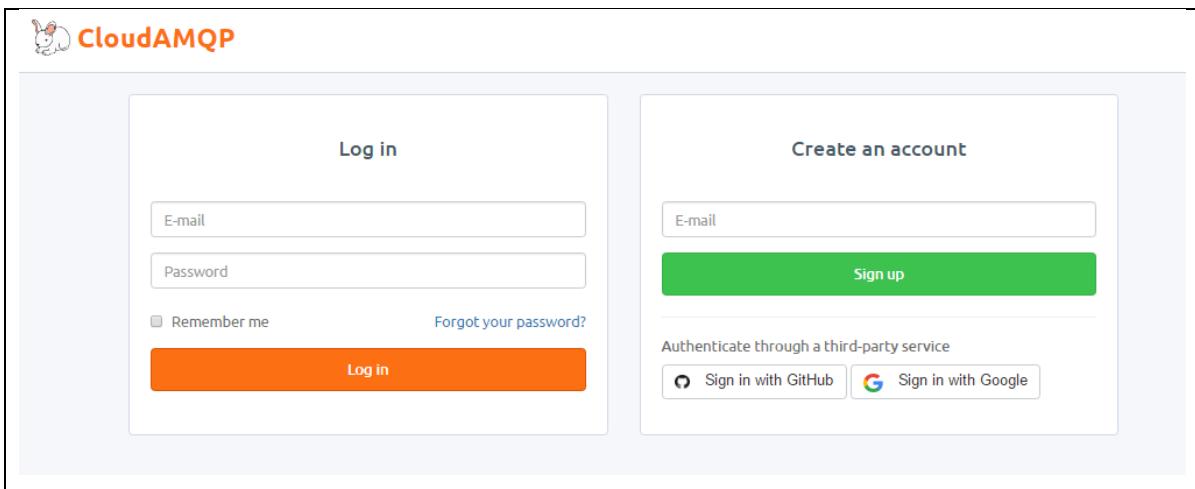
CloudAMQP digunakan sebagai messaging server online. Supaya dapat menggunakan CloudAMQP, dilakukan pendaftaran akun terlebih dahulu. Pendaftaran CloudAMQP dapat dilakukan dengan membuka tautan dibawah ini pada browser internet.

<https://www.cloudamqp.com/plans.html>

Apabila menelaah halaman web ini, disediakan beberapa plan untuk membuat sebuah instance pada CloudAMQP. Proyek ini menggunakan plan Little Lemur karena tidak berbayar. Klik tombol “Try Little Lemur now” untuk menggunakan plan ini.



Kemudian akan terbuka halaman web untuk melakukan login atau pendaftaran. Pendaftaran bisa dilakukan dengan melakukan sign up atau dengan menggunakan akun Github maupun akun Google.



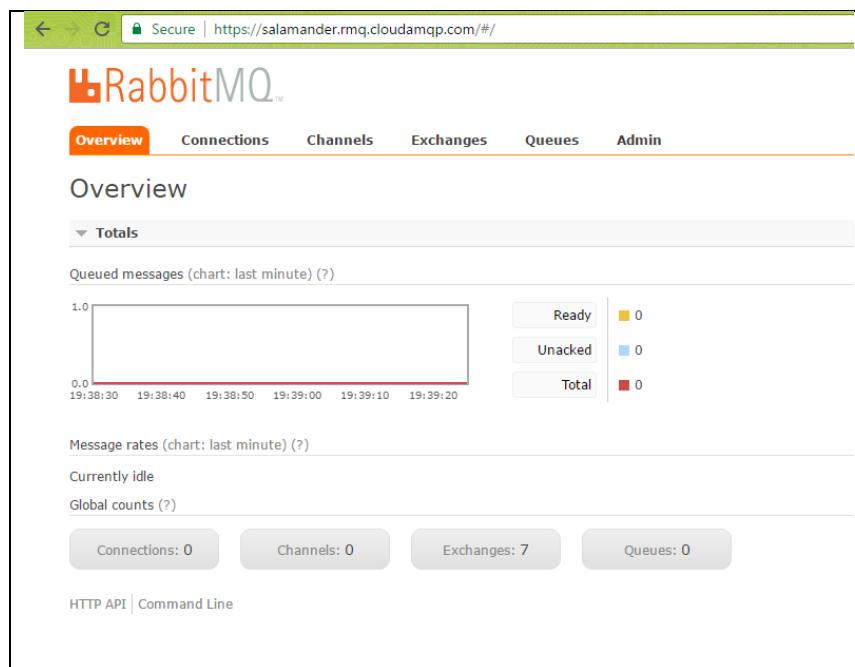
Setelah melakukan pendaftaran, maka akan terbuka halaman web untuk membuat sebuah instance sesuai dengan plan yang dipilih. Masukan nama Instance yang diinginkan, plan yang digunakan, dan data center. Data center dapat dipilih dengan bebas namun disarankan dekat dengan negara tempat mengakses server agar dapat mengurangi latency.

The screenshot shows the 'Edit instance' configuration screen. It includes a message about credit cards, fields for 'Name' (Earthquake Detector), 'Plan' (Little Lemur (Free)), and 'Data center' (US-East-1 (Northern Virginia)). It features an 'Amazon Web Services' logo and a 'Save Changes' button. To the right, there is a 'Plan' section with an illustration of a 'Little Lemur' and a link to the 'plan page'. A note at the bottom right says to see the 'plan page' for more information.

Akan muncul sebuah list instance ketika membuat instance lebih dari sekali. Artinya, ada beberapa server yang bisa digunakan sebagai messaging server. Namun, server tidak berbayar ini memiliki batasan yaitu jumlah message dan koneksi yang terbatas. Apabila hendak mengatur konfigurasi server, maka dapat dilakukan dengan klik tombol ‘RabbitMQ Manager’ di sebelah kanan layar browser.

Instances			
Name	Plan	Datacenter	Actions
Earthquake Detector	Lemur	Amazon Web Services US-East-1 (Northern Virginia)	Edit RabbitMQ Manager
PythonTest2	Lemur	Amazon Web Services US-East-1 (Northern Virginia)	Edit RabbitMQ Manager
PythonTest	Lemur	Google Compute Engine Europe West	Edit RabbitMQ Manager

Setelah itu, pada browser akan terbuka halaman baru yang merujuk pada alamat server. Ketika pertama kali membuka halaman web server ini, maka tampilan yang akan dilihat adalah sebagai berikut:



Hal yang dilakukan pertama kali adalah dengan mengklik tab Queues pada bagian atas layar.

The screenshot shows the RabbitMQ Management Console interface. The top navigation bar includes tabs for Overview, Connections, Channels, Exchanges, Queues (which is highlighted in orange), and Admin. Below the tabs, the main content area is titled "Queues". A sub-section titled "All queues" is displayed, showing a single entry "... no queues ...". Below this, a form for "Add a new queue" is present. The "Name" field contains "ecn" (highlighted in yellow). The "Durability" dropdown is set to "Durable". The "Auto delete" dropdown is set to "No". An "Arguments" section is shown with a key-value pair where the key is empty and the value is "String". Below the arguments, there are several optional configuration links: Message TTL (?), Auto expire (?), Max length (?), Max length bytes (?), Dead letter exchange (?), Dead letter routing key (?), and Maximum priority (?). At the bottom of the form is a blue "Add queue" button.

Pada tab Queues akan terdapat kolom name, durability, auto delete, dsb. Kolom name diisi dengan nama Queue (dalam proyek ini, menggunakan nama “ecn”), kolom durability diisi dengan pilihan “Durable”, sementara itu bagian lain tidak perlu diubah. Setelah itu, tekan tombol Add queue di bagian bawah halaman. Apabila sudah menekan tombol tersebut, maka akan muncul sebuah list queue dengan judul “ecn”.

The screenshot shows the RabbitMQ Management Console interface, specifically the Queues tab. The main content area is titled "Queues". A sub-section titled "All queues" is displayed, showing a table with one row for the queue "ecn". The table has columns for Name, Features, State, Ready, Unacked, Total, incoming, deliver / get, and ack. The "ecn" row shows "D HA" under Features, "idle" under State, and "0" for all other metrics. To the right of the table is a "+/-" button.

Kemudian hal selanjutnya yang harus dilakukan adalah klik tab Exchanges di bagian atas halaman.

Exchanges					
▼ All exchanges					
Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	D			
amq.headers	headers	D			
amq.match	headers	D			
amq.rabbitmq.trace	topic	D I			
amq.topic	topic	D			

Akan ditampilkan layar exchanges pada halaman web. Klik kalimat amq.topic pada bagian bawah list exchange.

Add binding from this exchange

To queue	:	ecn	*	
Routing key:				earthquake
Arguments:			=	String
<input type="button" value="Bind"/>				

Kemudian dilakukan binding exchange dengan queue. Isi kolom seperti gambar diatas. Hal ini dilakukan supaya queue dihubungkan dengan exchange melalui jalur routing key “earthquake”.

Apabila sudah melakukan bindings, maka akan terdapat tampilan seperti dibawah ini

▼ Bindings			
This exchange			
↓			
To	Routing key	Arguments	
ecn	earthquake		<input type="button" value="Unbind"/>

Konfigurasi CloudAMQP sebagai messaging server online bisa dikatakan cukup hingga tahap ini. Penggunaan server ini pada beberapa devais harus mengacu pada syntax yang telah disediakan oleh dokumentasi RabbitMQ sehingga akan diperoleh konfigurasi berbeda pada platform yang berbeda.

2.1.4.2 Implementasi RabbitMQ dengan menggunakan server RabbitMQ PPTIK

Pada proyek ini, produk akhir harus menggunakan server dari PPTIK ITB. Oleh karena itu dilakukan beberapa pengaturan yang akan dijabarkan berikut ini.

RabbitMQ diimplementasikan dengan menggunakan protokol AMQP seperti dibawah ini:

```
amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster
```

URL diatas merupakan merupakan parameter yang harus ditambahkan pada proses koneksi pengiriman dan pengambilan data dat

Keterangannya adalah sebagai berikut:

- a. amqp → Protokol komunikasi
- b. sensor_gempa → username server
- c. 12345 → password
- d. 167.205.7.226 → host server
- e. /%2fdisaster → /disaster → virtual host

Penggunaan %2f disini dimaksudkan untuk mengkoreksi kesalahan pembacaan /disaster karena karakter / berpengaruh terhadap protokol komunikasi dalam bentuk URL. Kemudian selain mengatur alamat server seperti yang dijelaskan diatas, pengaturan berikut-berikutnya akan dijelaskan sebagai berikut:

- a. queue → ecn + nomer sensor
penamaan queue dibebaskan namun harus unik dan khusus dialokasikan untuk sebuah sensor.
- b. exchange → amq.topic
amq.topic merupakan sebuah exchange default dengan type = topic yang dapat dipakai dalam proses implementasi
- c. routing key → exchange + queue
routing key dimaksudkan untuk menghubungkan exchange menuju queue tertentu. Hal ini bertujuan untuk memilih pesan sesuai dengan jenis dan nama sensornya.
- d. durable → true
durable bertujuan untuk menahan pesan supaya tidak hilang ketika server mati.

Sehingga implementasi untuk parameter-parameter diatas adalah sebagai berikut:

```
factory.Uri = "amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster";
channel.QueueDeclare(queue: "ecn",
                     durable: true,
                     exclusive: false,
                     autoDelete: false,
```

```

        arguments: null);
channel.QueueBind(queue: "ecn",
                  exchange: "amq.topic",
                  routingKey: "amq.topic.ecn"
                );

```

Penggunaan kode diatas adalah implementasi dalam Bahasa C#.

2.1.5 Implementasi Database MySQL

Supaya dapat memasukkan pesan kedalam MySQL, maka terlebih dahulu data di consume dengan menggunakan sebuah program. Program yang akan dibuat adalah program C# yang bertujuan untuk melakukan consume pada RabbitMQ dan mendapatkan pesan. Lalu pesan ini dipilah-pilah sesuai dengan kolom yang ada pada MySQL. Algoritma dari program tersebut adalah sebagai berikut:

```

public static void Main(string[] args)
{
    connect_database();
    Timer t = new Timer(TimerCallback, null, 0, 1000);
    Console.ReadLine();
}

```

Program diatas adalah program utama yang menjalankan fungsi koneksi database MySQL dan kemudian membuat Thread baru untuk menjalankan Timer dimana fungsi tersebut akan dipanggil setiap 1 detik.

```

public static void connect_msgserver()
{
    ConnectionFactory factory;
    using (StreamReader r = new StreamReader("config1.json"))
    {
        string json = r.ReadToEnd();
        Config config = JsonConvert.DeserializeObject<Config>(json);

        factory = new ConnectionFactory();
        factory.Uri = "amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster";
    }

    factory.Protocol = Protocols.DefaultProtocol;
    factory.Port = AmqpTcpEndpoint.UseDefaultPort;

    //SENSOR ecn
    using (var connection = factory.CreateConnection())
    using (var channel = connection.CreateModel())
    {

        channel.BasicQos(0, 1, false);
        channel.QueueDeclare(queue: "ecn",
                             durable: true,
                             exclusive: false,
                             autoDelete: false,
                             arguments: null);
        channel.QueueBind(queue: "ecn",
                          exchange: "amq.topic",
                          routingKey: "amq.topic.ecn"
                        );

        Console.WriteLine("Queue Declare Emergency GUI");
        var consumer = new EventingBasicConsumer(channel);
        consumer.Received += (model, ea) =>
        {
            var body = ea.Body;
            var message = Encoding.UTF8.GetString(body);
            try
            {
                data accelReport = JsonConvert.DeserializeObject<data>(message);
            }
            catch (Exception ex)
        };
    };
}

```

```

        {
            Console.WriteLine("Failed");
        }

    };
    channel.BasicConsume(queue: "ecn", //emergency_gui",
        noAck: true,
        consumer: consumer);

    Console.WriteLine("Already BasicConsume");
}
}

```

Program diatas adalah sebuah fungsi untuk melakukan consume pada RabbitMQ.

```

public static void connect_database()
{
    string server_host = "localhost";
    string server_password = "MySQLRoot";
    string server_name = "root";

    try
    {
        con.ConnectionString = "server=" + server_host + ";user id=" + server_name +
"password=" + server_password + ";database=earthquake";
        con.Open();
        //MessageBox.Show("Connected to " + server);

    }
    catch (Exception e1)
    {
        Console.WriteLine("Connection failed due to " + e1.ToString());
    }
}

private static void write_database(string[] data)
{
    command_add = con.CreateCommand();
    try
    {
        command_add.CommandText = "INSERT INTO store_id (point_time, time_zone_id,
interval_id, client_id, lattitude_id, longitude_id) VALUES('" + data[0] + "', '" + data[1] + "' ,
'" + data[2] + "','" + data[3] + "','" + data[4] + "','" + data[5] + "')";
        command_add.ExecuteNonQuery();
    }
    catch (Exception e1)
    {
        try
        {
            command_add.CommandText = "INSERT INTO store_id (point_time, time_zone_id,
interval_id, client_id, lattitude_id, longitude_id) VALUES('timestamp', '' ,'' + empty +
'', '' + empty + ',' + empty + ',' + empty + ',' + empty + ')";
            command_add.ExecuteNonQuery();
        }
        catch
        {
        }
    }
}

try
{
    con.Close();
}
catch
{
    Thread.Sleep(2000);
    con.Close();
}
Thread.Sleep(1000);
}

```

Program diatas adalah proses untuk melakukan proses koneksi pada database MySQL dan kemudian menuliskan data pada kolom-kolom di MySQL.

2.1.6 Implementasi pengambilan pesan oleh Consumer

Consumer yang dimaksud disini adalah GUI yang dirancang dengan menggunakan Microsoft Visual Studio 2013. Supaya GUI dapat mengambil pesan dari messaging server, maka diberikan algoritma pada seperti berikut:

```
ConnectionFactory factory;
    using (StreamReader r = new StreamReader("config1.json"))
    {
        string json = r.ReadToEnd();
        Config config = JsonConvert.DeserializeObject<Config>(json);

        factory = new ConnectionFactory{Uri = config.url};
    }

factory.Protocol = Protocols.DefaultProtocol;
factory.Port = AmqpTcpEndpoint.UseDefaultPort;
using (var connection = factory.CreateConnection())
using (var channel = connection.CreateModel())
{
    channel.QueueDeclare(queue: "ecn",
        durable: true,
        exclusive: false,
        autoDelete: false,
        arguments: null);
    channel.QueueBind(queue: "ecn",
        exchange: "amq.topic",
        routingKey: "amq.topic.ecn"
    );

    Console.WriteLine("Queue Declare Emergency GUI");
    var consumer = new EventingBasicConsumer(channel);
    consumer.Received += (model, ea) =>
    {
        var body = ea.Body;
        var message = Encoding.UTF8.GetString(body);
        Console.WriteLine($" [x] Received {0}", message);
        Console.WriteLine("-----");
        data = message;
        try
        {
            data accelReport =
JsonConvert.DeserializeObject<data>(message);
            Console.WriteLine("{0} {1} {2} {3} {4}",
accelReport.geojson.geometry.coordinates[0],
accelReport.geojson.geometry.coordinates[1], accelReport.accelerations[0].x,
accelReport.accelerations[0].y, accelReport.accelerations[0].z);
            for (int i = 0 ; i < 20; i++)
            {

perfChart.AddValue((decimal)int.Parse(accelReport.accelerations[i].x));
perfChart1.AddValue((decimal)int.Parse(accelReport.accelerations[i].y));
perfChart2.AddValue((decimal)int.Parse(accelReport.accelerations[i].z));
        }
        gMapControl1.Position = new
GMap.NET.PointLatLng(double.Parse(accelReport.geojson.geometry.coordinates[0]),
double.Parse(accelReport.geojson.geometry.coordinates[1])));
    }
}
```

```

        }
        catch (Exception ex)
        {
            Console.WriteLine("ERROR: {0}", ex);
        }

    };
    channel.BasicConsume(queue: "ecn", //emergency_gui",
                          noAck: true,
                          consumer: consumer);

    Console.WriteLine("Already BasicConsume");
}

```

Dalam algoritma tersebut, terdapat syntax yang bertugas membaca file json dalam suatu directory yang serupa dengan directory GUI. Setelah membaca file json tersebut, akan diambil isi dari file yang bertugas untuk mengakses sebuah messaging server.

```

using (StreamReader r = new StreamReader("config1.json"))
{
    string json = r.ReadToEnd();
    Config config = JsonConvert.DeserializeObject<Config>(json);

    factory = new ConnectionFactory{Uri = config.url};
}

```

Kemudian dilakukan akses pada queue yang terdapat pada messaging server.

```

using (var connection = factory.CreateConnection())
using (var channel = connection.CreateModel())
{
    channel.QueueDeclare(queue: "ecn",
                          durable: true,
                          exclusive: false,
                          autoDelete: false,
                          arguments: null);
    channel.QueueBind(queue: "ecn",
                      exchange: "amq.topic",
                      routingKey: "amq.topic.ecn"
    );
}

```

Setelah itu, dilakukan proses consume data pada queue yang berkaitan.

```

var consumer = new EventingBasicConsumer(channel);
consumer.Received += (model, ea) =>
{
    var body = ea.Body;
    var message = Encoding.UTF8.GetString(body);
    ...
}

```

Kemudian dilakukan parsing dengan menambahkan syntax berikut ini:

```
data accelReport = JsonConvert.DeserializeObject<data>(message);
```

Data merupakan sebuah tipe data object dengan deklarasi seperti ini:

```

class data
{
    public string pointTime;
}

```

```

        public string timeZone;
        public string interval;
        public geojson geojson;
        public acc[] accelerations;
    };

    class geojson
    {
        public string type;
        public geometry geometry;
        public prop property;
    };

    class geometry
    {
        public string type;
        public string[] coordinates;
    };

    class prop
    {
        public string name;
    };

    class acc
    {
        public string x;
        public string y;
        public string z;
    };
}

```

Lampiran Program Database.cs

```

using MySql.Data.MySqlClient;
using Newtonsoft.Json;
using RabbitMQ.Client;
using RabbitMQ.Client.Events;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Database
{
    public static class Program
    {
        public static string timestamp = "";
        public static string empty = "-";
        public static string[] words1 = new string[2];
        public static string[] words = new string[10];
        public static string[] messages;
        public static int count = 0;
        public static int count1 = 0;
        public static MySqlCommand command_add;
        public static MySqlConnection con = new MySqlConnection();

        public static void Main(string[] args)
        {
            connect_database();
            Timer t = new Timer(TimerCallback, null, 0, 1000);
            Console.ReadLine();
        }
    }
}

```

```

public static void connect_msgserver()
{
    ConnectionFactory factory;
    using (StreamReader r = new StreamReader("config1.json"))
    {
        string json = r.ReadToEnd();
        Config config = JsonConvert.DeserializeObject<Config>(json);

        factory = new ConnectionFactory();
        factory.Uri =
"amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster";
    }

    factory.Protocol = Protocols.DefaultProtocol;
    factory.Port = AmqpTcpEndpoint.UseDefaultPort;

    //SENSOR ecn
    using (var connection = factory.CreateConnection())
    using (var channel = connection.CreateModel())
    {

        channel.BasicQos(0, 1, false);
        channel.QueueDeclare(queue: "ecn",
            durable: true,
            exclusive: false,
            autoDelete: false,
            arguments: null);
        channel.QueueBind(queue: "ecn",
            exchange: "amq.topic",
            routingKey: "amq.topic.ecn"
        );

        Console.WriteLine("Queue Declare Emergency GUI");
        var consumer = new EventingBasicConsumer(channel);
        consumer.Received += (model, ea) =>
        {

            var body = ea.Body;
            var message = Encoding.UTF8.GetString(body);
            try
            {
                data accelReport =
JsonConvert.DeserializeObject<data>(message);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Failed");
            }

        };
        channel.BasicConsume(queue: "ecn", //"emergency_gui",
            noAck: true,
            consumer: consumer);

        Console.WriteLine("Already BasicConsume");
    }
}

public static void connect_msgserver2()
{
    //consume_data();
    ConnectionFactory factory;
    factory = new ConnectionFactory();
    factory.Uri =
"amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster";
    factory.Protocol = Protocols.DefaultProtocol;
    factory.Port = AmqpTcpEndpoint.UseDefaultPort;
}

```

```

//SENSOR ecn
using (var connection = factory.CreateConnection())
using (var channel = connection.CreateModel())
{
    channel.BasicQos(0, 1, false);
    channel.QueueDeclare(queue: "ecn", //"emergency_gui",
        durable: true,
        exclusive: false,
        autoDelete: false,
        arguments: null);
    channel.QueueBind(queue: "ecn", //"emergency_gui",
        exchange: "amq.topic",
        routingKey: "amq.topic.ecn" //emergency
    );

    Console.WriteLine("Queue Declare Emergency GUI");
    var consumer = new EventingBasicConsumer(channel);
    consumer.Received += (model, ea) =>
    {
        var body = ea.Body;
        var message = Encoding.UTF8.GetString(body);
        try
        {
            words1[1] = message;
            Console.WriteLine(message);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Failed");
        }
    };
    channel.BasicConsume(queue: "ecn", //"emergency_gui",
        noAck: true,
        consumer: consumer);

    Console.WriteLine("Already BasicConsume");
    //Console.ReadLine();
}
}

public static void connect_database()
{
    string server_host = "localhost";
    string server_password = "MySQLRoot";
    string server_name = "root";

    try
    {
        con.ConnectionString = "server=" + server_host + ";user id=" +
server_name + ";password=" + server_password + ";database=earthquake";
        con.Open();
        //MessageBox.Show("Connected to " + server);

    }
    catch (Exception e1)
    {
        Console.WriteLine("Connection failed due to " +
e1.ToString());
    }
}

private static void write_database(string[] data)
{
    command_add = con.CreateCommand();
    try

```

```

        {
            command_add.CommandText = "INSERT INTO store_id (point_time,
time_zone_id, interval_id, client_id, lattitude_id, longitude_id) VALUES('" +
data[0] + "'", "" + data[1] + "", "" + data[2] + "", "" + data[3] + "", "" +
data[4] + "", "" + data[5] + "")";
            command_add.ExecuteNonQuery();
        }
        catch (Exception e1)
        {
            try
            {
                command_add.CommandText = "INSERT INTO store_id
(point_time, time_zone_id, interval_id, client_id, lattitude_id, longitude_id)
VALUES('" + timestamp + "', '" + empty + "','" + empty + "','" + empty + "','" +
empty + "','" + empty + "')";
                command_add.ExecuteNonQuery();
            }
            catch
            {
            }
        }

        try
        {
            con.Close();
        }
        catch
        {
            Thread.Sleep(2000);
            con.Close();
        }
        Thread.Sleep(1000);
    }

    private static void write_database_test(string[] data)
    {
        command_add = con.CreateCommand();
        try
        {
            command_add.CommandText = "INSERT INTO test_id
(point_time_console, point_time_message) VALUES('" + data[0] + "', '" +
data[1] + "')";
            command_add.ExecuteNonQuery();
        }
        catch (Exception e1)
        {
            try
            {
                command_add.CommandText = "INSERT INTO test_id
(point_time_console, point_time_message) VALUES('" + timestamp + "', '" + empty +
+ "')";
                command_add.ExecuteNonQuery();
            }
            catch
            {
            }
        }

        //Reset connection to avoid exception
        try
        {
            con.Close();
        }
        catch
        {
            Thread.Sleep(2000);
        }
    }
}

```

```

        con.Close();
    }

    try
    {
        con.Open();
    }
    catch
    {
        Thread.Sleep(2000);
        con.Close();
    }

    Thread.Sleep(1000);

}

private static void TimerCallback(Object o)
{
    words[0] = "";
    words1[0] = "";
    // Display the date/time when this method got called.
    //Console.WriteLine("In TimerCallback: " +
DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));
    //connect_msgserver();
    timestamp = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
    words[0] = timestamp;
    words1[0] = timestamp;
    connect_msgserver2();
    //connect_msgserver2();
    Console.WriteLine(timestamp);
    write_database_test(words1);
    //write_database_test(words1);
    //count++;
    // Force a garbage collection to occur for this demo.
    GC.Collect();
}

private static void TimerCallback1(Object o)
{
    GC.Collect();
}

public class Config
{
    public string host;
    public string user;
    public string vhost;
    //public string port;
    public string password;
}

}

public class data
{
    public string pointTime;
    public string timeZone;
    public string interval;
    public string clientID;
    public geojson geojson;
    public acc[] accelerations;
};

public class geojson

```

```

{
    public geometry geometry;
    public prop property;
};

public class geometry
{
    public string type;
    public string[] coordinates;
};

public class prop
{
    public string name;
};

public class acc
{
    public string x;
    public string y;
    public string z;
};

}

using MySql.Data.MySqlClient;
using Newtonsoft.Json;
using RabbitMQ.Client;
using RabbitMQ.Client.Events;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Database
{
    public static class Program
    {
        public static string timestamp = "";
        public static string empty = "-";
        public static string[] words1 = new string[2];
        public static string[] words = new string[10];
        public static string[] messages;
        public static int count = 0;
        public static int count1 = 0;
        public static MySqlCommand command_add;
        public static MySqlConnection con = new MySqlConnection();

        public static void Main(string[] args)
        {
            connect_database();
            Timer t = new Timer(TimerCallback, null, 0, 1000);
            Console.ReadLine();
        }

        public static void connect_msgserver()
        {
            ConnectionFactory factory;
            using (StreamReader r = new StreamReader("config1.json"))
            {
                string json = r.ReadToEnd();
                Config config = JsonConvert.DeserializeObject<Config>(json);

                factory = new ConnectionFactory();
                factory.Uri =
                    "amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster";
            }

            factory.Protocol = Protocols.DefaultProtocol;
            factory.Port = AmqpTcpEndpoint.UseDefaultPort;
        }
    }
}

```

```

//SENSOR ecn
using (var connection = factory.CreateConnection())
using (var channel = connection.CreateModel())
{
    channel.BasicQos(0, 1, false);
    channel.QueueDeclare(queue: "ecn",
        durable: true,
        exclusive: false,
        autoDelete: false,
        arguments: null);
    channel.QueueBind(queue: "ecn",
        exchange: "amq.topic",
        routingKey: "amq.topic.ecn"
    );

    Console.WriteLine("Queue Declare Emergency GUI");
    var consumer = new EventingBasicConsumer(channel);
    consumer.Received += (model, ea) =>
    {
        var body = ea.Body;
        var message = Encoding.UTF8.GetString(body);
        try
        {
            data accelReport =
JsonConvert.DeserializeObject<data>(message);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Failed");
        }
    };
    channel.BasicConsume(queue: "ecn", //emergency_gui",
        noAck: true,
        consumer: consumer);

    Console.WriteLine("Already BasicConsume");
}
}

public static void connect_msgserver2()
{
    //consume_data();
    ConnectionFactory factory;
    factory = new ConnectionFactory();
    factory.Uri =
"amqp://sensor_gempa:12345@167.205.7.226/%2fdisaster";
    factory.Protocol = Protocols.DefaultProtocol;
    factory.Port = AmqpTcpEndpoint.UseDefaultPort;

    //SENSOR ecn
    using (var connection = factory.CreateConnection())
    using (var channel = connection.CreateModel())
    {

        channel.BasicQos(0, 1, false);
        channel.QueueDeclare(queue: "ecn", //emergency_gui",
            durable: true,
            exclusive: false,
            autoDelete: false,
            arguments: null);
        channel.QueueBind(queue: "ecn", //emergency_gui",
            exchange: "amq.topic",
            routingKey: "amq.topic.ecn" //emergency
        );
    }
}

```

```

Console.WriteLine("Queue Declare Emergency GUI");
var consumer = new EventingBasicConsumer(channel);
consumer.Received += (model, ea) =>
{
    var body = ea.Body;
    var message = Encoding.UTF8.GetString(body);
    try
    {
        words1[1] = message;
        Console.WriteLine(message);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Failed");
    }
};

channel.BasicConsume(queue: "ecn", // "emergency_gui",
                      noAck: true,
                      consumer: consumer);

Console.WriteLine("Already BasicConsume");
//Console.ReadLine();
}
}

public static void connect_database()
{
    string server_host = "localhost";
    string server_password = "MySQLRoot";
    string server_name = "root";

    try
    {
        con.ConnectionString = "server=" + server_host + ";user id=" +
server_name + ";password=" + server_password + ";database=earthquake";
        con.Open();
        //MessageBox.Show("Connected to " + server);
    }
    catch (Exception e1)
    {
        Console.WriteLine("Connection failed due to " +
e1.ToString());
    }
}

private static void write_database(string[] data)
{
    command_add = con.CreateCommand();
    try
    {
        command_add.CommandText = "INSERT INTO store_id (point_time,
time_zone_id, interval_id, client_id, lattitude_id, longitude_id) VALUES('" +
data[0] + "'", "' + data[1] + '", "' + data[2] + '", "' + data[3] + '", "' +
data[4] + "'", "' + data[5] + "')";
        command_add.ExecuteNonQuery();
    }
    catch (Exception e1)
    {
        try
        {
            command_add.CommandText = "INSERT INTO store_id
(point_time, time_zone_id, interval_id, client_id, lattitude_id, longitude_id)
VALUES('" + timestamp + "','" + empty + "','" + empty + "','" + empty + "','" +
empty + "','" + empty + "')";
        
```

```

        command_add.ExecuteNonQuery();
    }
    catch
    {
    }

}

try
{
    con.Close();
}
catch
{
    Thread.Sleep(2000);
    con.Close();
}
Thread.Sleep(1000);
}

private static void write_database_test(string[] data)
{
    command_add = con.CreateCommand();
    try
    {
        command_add.CommandText = "INSERT INTO test_id
(point_time_console, point_time_message) VALUES('" + data[0] + "', '" +
data[1] + "')";
        command_add.ExecuteNonQuery();
    }
    catch (Exception e1)
    {
        try
        {
            command_add.CommandText = "INSERT INTO test_id
(point_time_console, point_time_message) VALUES('" + timestamp + "','" + empty +
"'')";
            command_add.ExecuteNonQuery();
        }
        catch
        {
        }
    }
}

//Reset connection to avoid exception
try
{
    con.Close();
}
catch
{
    Thread.Sleep(2000);
    con.Close();
}

try
{
    con.Open();
}
catch
{
    Thread.Sleep(2000);
    con.Close();
}

Thread.Sleep(1000);

```

```

        }

    private static void TimerCallback(Object o)
    {
        words[0] = "";
        words1[0] = "";
        // Display the date/time when this method got called.
        //Console.WriteLine("In TimerCallback: " +
DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));
        //connect_msgserver();
        timestamp = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
        words[0] = timestamp;
        words1[0] = timestamp;
        connect_msgserver2();
        //connect_msgserver2();
        Console.WriteLine(timestamp);
        write_database_test(words1);
        //write_database_test(words1);
        //count++;
        // Force a garbage collection to occur for this demo.
        GC.Collect();
    }

private static void TimerCallback1(Object o)
{
    GC.Collect();
}

public class Config
{
    public string host;
    public string user;
    public string vhost;
    //public string port;
    public string password;
}

}

public class data
{
    public string pointTime;
    public string timeZone;
    public string interval;
    public string clientID;
    public geojson geojson;
    public acc[] accelerations;
};

public class geojson
{
    public geometry geometry;
    public prop property;
};

public class geometry
{
    public string type;
    public string[] coordinates;
};

public class prop
{
    public string name;
};

public class acc
{
}

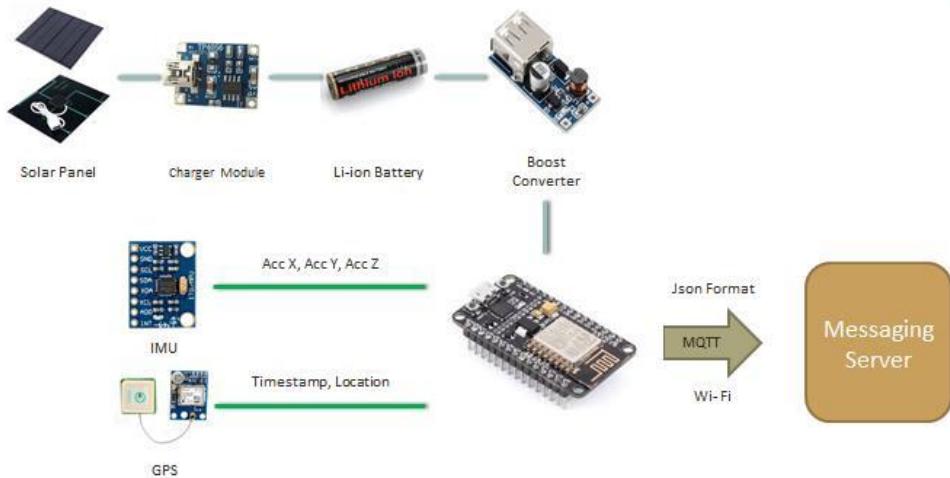
```

```

    public string x;
    public string y;
    public string z;
}
}

```

2.2 Sensor Earthquake Catcher Network



Gambar Diagram Blok Sensor ECN

Pada sistem yang kita buat, Sensor ECN berfungsi untuk membaca data seismik, timestamp, dan lokasi dari sensor tersebut. Seperti yang sudah dijabarkan pada dokumen-dokumen sebelumnya, sensor kami memiliki beberapa spesifikasi yang harus dipenuhi yaitu harga sensor yang jauh lebih rendah dari harga seismometer(total harga 3 juta), peforma sensor yang tidak terganggu dengan kondisi lingkungan sekitarnya, sensor dapat bekerja selama 24 jam tanpa henti, daya yang digunakan sensor rendah (<1W), dan akurasi dari pembacaan sensor getaran tinggi (ADC 16 bit).

Desain sensor ECN terdiri dari tiga bagian, yaitu modul utama dan modul charge controller. Modul utama berfungsi untuk membaca data-data yang diperlukan seperti data getaran (percepatan), Timestamp, dan Lokasi dari sensor dan mengirimkannya menggunakan protokol MQTT ke Messaging server (cloudAMQP) dengan menggunakan Wi-Fi, Modul Charge Controller berfungsi sebagai penyedia sumber daya yang terus-menerus bekerja untuk modul utama.

2.1.1 Main Module

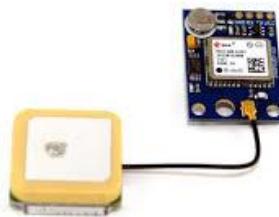
2.1.1.1 Elektrikal

Terdapat tiga komponen utama pada main module, yaitu NodeMCU v2, MPU9255, dan GPS Ublox Neo 6M.



Gambar NodeMCU v2

NodeMCU merupakan mikrokontroller yang sudah built-in modul Wi-Fi sehingga tidak diperlukan modul Wi-Fi tambahan. NodeMCU digunakan sebagai mikrokontroler yang bertugas untuk melakukan proses sampling data percepatan sebanyak 40 kali selama 1 detik, membaca data lokasi dan timestamp yang diberikan oleh GPS, lalu mengirimkan data-data tersebut ke server dengan menggunakan protokol MQTT melalui jaringan Wi-Fi.



Gambar GPS Ublox Neo 6M

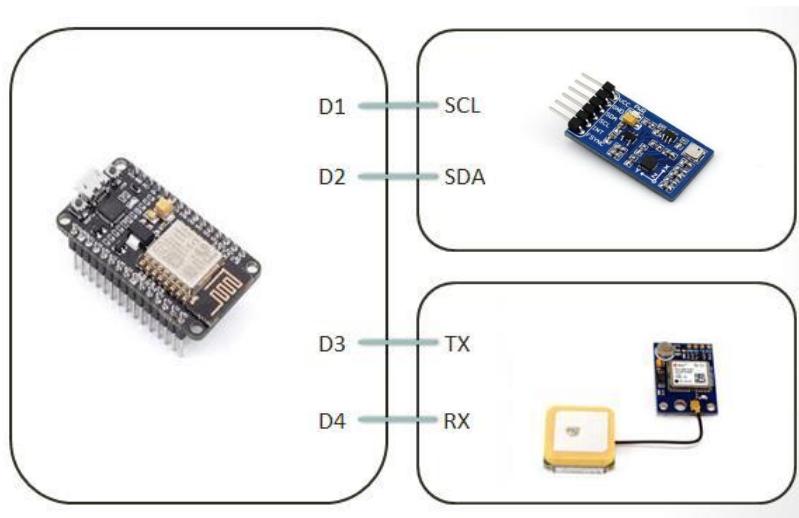
Untuk mendapatkan lokasi sensor dan data waktu yang dapat sinkron dengan sensor-sensor yang lain, digunakan modul GPS Ublox Neo 6M. Modul GPS ini terus menerus mendapat informasi dari satelit dan informasi-informasi tersebut dapat diperoleh NodeMCU dengan menggunakan komunikasi serial. Gelombang yang diterima oleh antena GPS ini memiliki daya tembus yang rendah sehingga modul GPS ini harus diletakkan di tempat yang tidak terhalang oleh tembok/atap. Data-data yang dapat diperoleh dari modul GPS ini adalah data latitude dan longitude serta data-data timestamp yang tersinkronisasi seperti detik, menit, jam, tanggal, bulan, dan tahun



Gambar Sensor IMU

Modul yang digunakan untuk mendapatkan data getaran adalah modul MPU9255. Pada modul MPU9255 ini terdapat accelerometer sehingga dapat diperoleh data percepatan dari getaran. Komunikasi yang digunakan modul MPU9255 ini adalah I²C. Inisisasi pembacaan data dari MPU9255 dilakukan dengan cara wake-up MPU9255 atau dengan

menuliskan bit 0 ke register 0x6B. Setelah dilakukan inisiasi, data percepatan x, y, dan z dapat diperoleh dengan membaca register 0x3B sampai 0x3F.

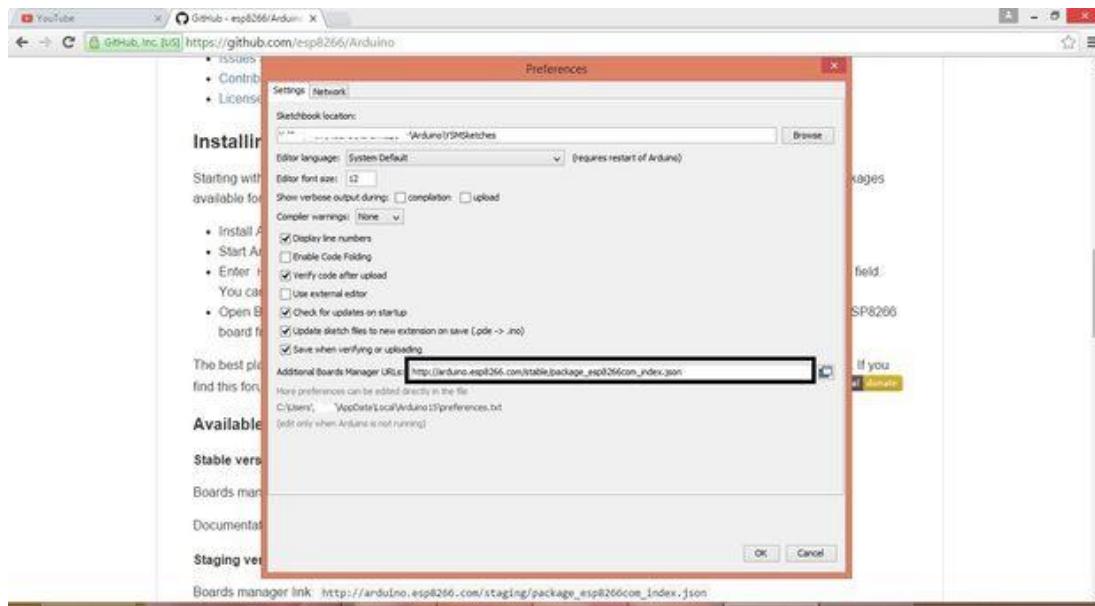


Gambar Skematik Modul Utama Sensor ECN

Untuk interkoneksi antar komponen dapat dilihat pada gambar diatas. Pin SCL pada MPU9255 dihubungkan dengan pin D1 pada NodeMCU dan pin SDA pada MPU9255 dihubungkan dengan pin D2 pada NodeMCU agar komunikasi I²C dapat dijalankan. Sedangkan pin Tx pada GPS dihubungkan dengan pin D3 pada NodeMCU dan pin Rx pada GPS dihubungkan dengan pin D4 pada NodeMCU. Pin D3 dan D4 merupakan pin yang dibuat menjadi pin Rx dan Tx dengan menggunakan software serial. Tidak digunakannya pin Rx dan Tx yang sudah ada pada NodeMCU karena pin Rx dan Tx tersebut sudah digunakan untuk serial monitor sehingga proses debugging lebih mudah dilakukan.

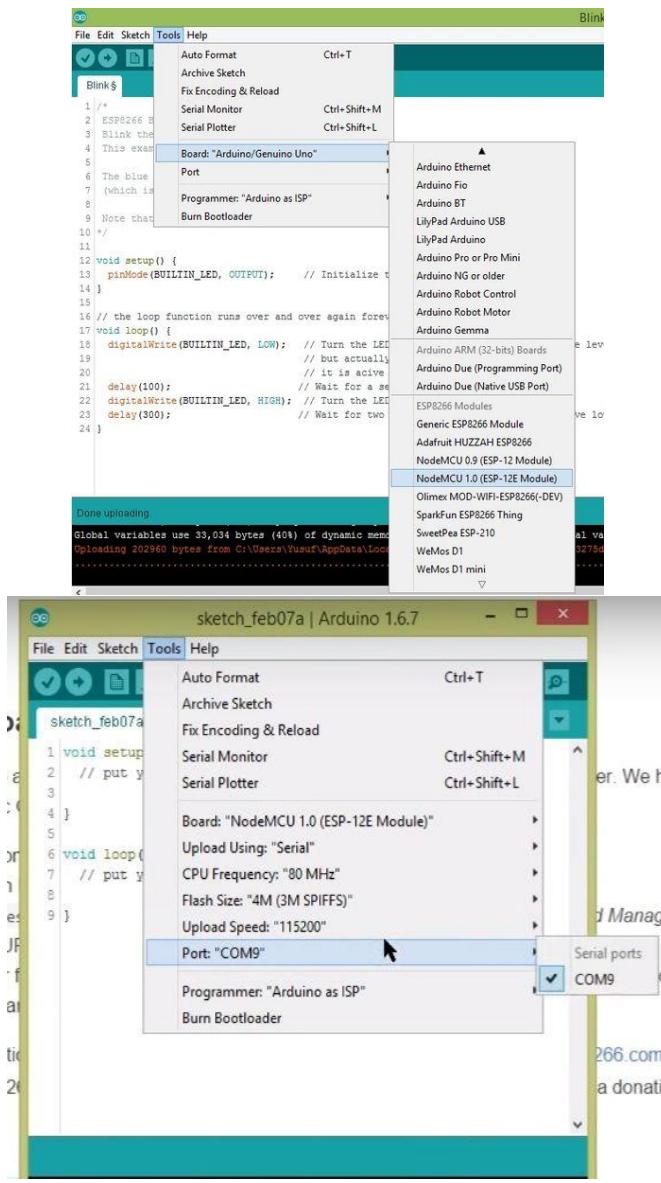
2.1.1.2 Algoritma

Digunakan IDE Arduino untuk mempermudah proses debugging dan upload program ke NodeMCU. Langkah pertama yang dilakukan adalah pada bagian preferences pada IDE Arduino, kolom Additional Board Manager URLs diisi dengan http://arduino.esp8266.com/stable/package_esp8266com_index.json



Gambar Tampilan untuk Menambahkan Board pada IDE Arduino

Setelah itu IDE Arduino harus di-restart. Setelah IDE Arduino dibuka kembali, board NodeMCU sudah terinstall pada IDE Arduino. Langkah selanjutnya yang dilakukan adalah memilih board pada bagian Tools -> Board.



Gambar Tampilan untuk Setting Board dan Port

Setelah itu pilih Port pada bagian Tools -> Port. Setelah langkah-langkah tersebut dilakukan, proses kompilasi dan upload program ke NodeMCU dapat dilakukan pada IDE Arduino.

```

//=====
//=====Library & Constant=====
//=====

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <TinyGPS++.h>
#include <ArduinoJson.h>

#define SDA_PIN D1
#define SCL_PIN D2
#define PWR_MGMT 0x6B
#define RXPin D3
#define TXPin D4
#define GPSBaud 9600
#define NData 40 // Amount of Data per one second

const char* TIMEZONE = "Asia/Jakarta";
const char* PROP = "ITB";
const char* TYPE = "Point";
const String SensorID = "1";
const int SendPeriod = 1000; //in ms
const int WaitGPS = 10;
const int N = NData * (SendPeriod/1000); // Amount of Sample
//Initial Coordinate
static const double init lat = -6.889916, init lon = 107.61133;

```

Gambar Potongan Source Code Include Library & Deklarasi Konstanta

Pada algoritma awal, dilakukan beberapa inisiasi seperti include library dan pendefinisian konstanta. Pada sensor ECN ini digunakan 5 library yaitu library ESP8266Wifi untuk koneksi Wi-Fi pada NodeMCU, PubSubClient untuk protokol MQTT, SoftwareSerial untuk deklarasi pin GPIO sebagai pin Serial tambahan yang digunakan untuk modul GPS, Wire untuk komunikasi I²C yang digunakan untuk memperoleh data dari MPU9255, dan Library TinyGPS++ untuk pembacaan data yang diperoleh dari GPS. Selain itu dilakukan juga deklarasi pin yang digunakan untuk komunikasi I²C dan Serial serta deklarasi register-register pada MPU9255. Konstanta GPSBaud merupakan baudrate yang digunakan untuk berkomunikasi serial dengan modul GPS. Konstanta SendPeriod merupakan periode pengiriman message. Konstanta Ndata adalah jumlah data yang disample dalam waktu satu detik, diisi dengan nilai 40 yang menandakan bahwa frekuensi sampling adalah 40 Hz. Konstanta char TIMEZONE, PROP, TYPE merupakan inisialisasi nilai suatu objek Json. Konstanta SensorID mengimplikasikan bahwa sensor tersebut memiliki ID 1, SensorID ini diperlukan untuk pembeda identitas antar sensor ketika data berada pada server. SendPeriod menandakan bahwa sensor akan mengirimkan data ke server setiap 1 detik. WaitGPS untuk menentukan seberapa lama sensor akan menunggu GPS sampai GPS dapat mengunci komunikasi dengan satelit. Konstanta N mengindikasikan jumlah sampling data yang dikirimkan ke server.

Jumlah data sampling dalam sekali kirim

$$= \text{Jumlah sample data dalam satu detik} * \left(\frac{\text{Periode pengiriman data ke server (ms)}}{1000 \text{ ms}} \right)$$

```

//=====
//=====Connection & Database Variables=====
//=====

const char* ssid = "LSKK Basement";      // network SSID (name)
const char* pass = "noiznocon";          // network password
const char* mqtt_server = "167.205.7.226";
const char* server_topic = "amq.topic.ecn"; //MQTT server topic
String mqtt_clientID = "ECN-" + SensorID;
String mqtt_user = "/disaster:sensor_gempa";
String mqtt_password = "12345";
int status = WL_IDLE_STATUS;
WiFiClient espClient;
PubSubClient client(espClient);

```

Gambar Potongan Source Code Deklarasi Variabel Koneksi dan MQTT

Untuk melakukan pengiriman diperlukan beberapa variabel yang harus digunakan. Untuk melakukan koneksi Wi-Fi dengan menggunakan NodeMCU hanya diperlukan dua variabel yaitu ssid dan pass yang bertipe array of constant character. Dua variabel ini menandakan access point mana yang ingin digunakan oleh NodeMCU. Sedangkan untuk pengaturan server MQTT digunakan variabel mqtt_server, server_topic, mqtt_clientID, mqtt_user, dan mqtt_password. Variabel mqtt_server digunakan untuk mendeklarasikan server manapun yang ingin digunakan pada CloudAMQP. Variabel server_topic digunakan untuk mendeklarasikan topic mana yang ingin dikirimkan message pada server. Format dari variabel ini adalah “x.t” dimana “x” merupakan nama exchange yang digunakan dan “t” merupakan nama topic yang digunakan pada CloudAMQP. Sedangkan variabel mqtt_clientID digunakan untuk mendeklarasikan nama client ketika melakukan koneksi ke server CloudAMQP. Nama yang digunakan boleh bebas asal tidak sama antara satu sensor dengan sensor yang lainnya. Variabel mqtt_user dan mqtt_password digunakan untuk mendeklarasikan username dari sensor beserta passwordnya. Format username yang digunakan adalah “v.u”, dimana “v” merupakan nama vhost yang digunakan dan “u” merupakan nama username yang digunakan. Dalam kasus dengan menggunakan CloudAMQP ini nama vhost dan username yang digunakan sama. Sedangkan espClient merupakan objek yang digunakan pada fungsi-fungsi pada library ESP8266WiFi dan objek PubSubClient merupakan hasil dari fungsi client pada library PubSubClient dengan input objek WiFiClient espClient.

```

//=====
//=====IMU & GPS Initiation=====
//=====

float temperaturG = 0.0;

struct MPU9255 {
    float x;
    float y;
    float z;
    float temp;
    float magx;
    float magy;
    float magz;
};

#define ACC_FULL_SCALE_2_G 0x00
#define ACC_FULL_SCALE_4_G 0x08
#define ACC_FULL_SCALE_8_G 0x10
#define ACC_FULL_SCALE_16_G 0x18

#define ACC_FILTER_OFF 0x08

const float ACC_RES = 6.1038817677e-05;
const float MAG_RES = 0.149540696432;
const float G = 9.8;

float axg = 0.0;
float ayg = 0.0;
float azg = 0.0;

float axgf = 0.0;
float aygf = 0.0;
float azgf = 0.0;

float mxt = 0.0;
float myt = 0.0;

MPU9255 data;
times t;

//TinyGPS++ Object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);

```

Gambar Potongan Source Code Deklarasi Variabel IMU

Pada bagian dari potongan source code yang digunakan diatas, dilakukan deklarasi beberapa variabel yang berhubungan dengan proses pembacaan data dari sensor IMU dan GPS. Konstanta MPU mengindikasikan alamat I²C dari MPU9255. Sedangkan struct MPU9255 dibuat untuk mempermudah parsing data ketika melakukan pembacaan dari sensor IMU. Lalu dilakukan juga inisiasi objek TinyGPSPlus dan SoftwareSerial yang digunakan pada library TinyGPS++ untuk proses pembacaan data yang diterima dari GPS. Struct times digunakan untuk menyimpan data millis() yang digunakan untuk kalkulasi lama proses dari suatu proses. Dapat dilihat juga didefinisikan beberapa register untuk memudahkan jika ingin mengganti konfigurasi penggunaan MPU9255. Konstanta float pada ACC_RES dan MAG_RES yang mengindikasikan resolusi pembacaan data percepatan dan magnetometer. Angka tersebut diperoleh dari persamaan sebagai berikut.

$$ACC_RES = \frac{2}{2^{16-1} - 1} = 6.10388817677e - 05$$

$$MAG_RES = \frac{4800}{2^{16-1} - 1} = 0.149540696432$$

Persamaan diatas merupakan persamaan untuk menghitung resolusi pembacaan accelerometer dan magnetometer. Nilai 2 & 4800 merupakan nilai maksimum dari pembacaan accelerometer dan magnetometer. Nilai tersebut dapat diubah-ubah dengan mengganti register pada MPU9255.

```
//=====
//=====JSON OBJECT=====
//=====
struct DataIMU {
    String x;
    String y;
    String z;
};

struct MessageData {
    String PointTime;
    String coordinates[2];
    DataIMU acc[N];
};

struct SensorSetting {
    String ClientID;
    String TimeZone;
    String Interval;
    String Properties;
};

struct MessageData payload_data;
struct SensorSetting payload_setting;
```

Gambar Potongan Source Code Deklarasi Objek Json

Dilakukan juga deklarasi objek yang berkorelasi dengan format message json yang ingin dikirimkan ke server. Format Json yang digunakan dapat dilihat dibawah ini.

```
{  
    "pointTime": "",  
    "timeZone": "Asia/Jakarta",  
    "interval": 1000,  
    "clientID": "ECN-1",  
    "geojson": {  
        "geometry": {  
            "type": "Point",  
            "coordinates": [125.6, 10.1]  
        },  
        "properties": {  
            "name": "ITB"  
        }  
    },  
    "accelerations": [  
        {  
            "x": -0.1243,  
            "y": 14.6464725,  
            "z": -12.5433  
        }  
    ]  
}
```

```

void setup()
{
    MPU9255_Init();
    payload_setting = InitJsonObject(payload_setting);
    WiFiConnect();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);

    Serial.begin(9600);
    Serial.println();
    ss.begin(GPSBaud);

    while (!checkgps)
    {
        while (ss.available() > 0)
        {
            if (gps.encode(ss.read()))
            {
                displayInfo();
                checkgps = true;
            }
        }
        if (millis() > WaitGPS*SendPeriod && gps.charsProcessed() < 10)
        {
            payload_data.coordinates[0] = String(init_lat);
            payload_data.coordinates[1] = String(init_lon);
            checkgps = true;
        }
    }
}

```

Gambar Potongan Source Code Setup

Pada bagian setup Arduino, dilakukan beberapa hal, yaitu inisiasi MPU9255 dan nilai awal objek Json. Setelah itu dilakukan percobaan untuk koneksi dengan access point Wi-Fi. Setelah itu, diatur juga server dan callback untuk protokol MQTT. Pada sensor ini tidak digunakan callback krn tidak menggunakan fitur subscribe MQTT. Setelah itu diinisiasi protokol pembacaan data dari GPS lalu mencoba untuk membaca data longitude dan latitude sampai batas waktu tertentu/konstanta WaitGPS. Ketika sudah berhasil diperoleh data longitude dan latitude, data tersebut disimpan pada objek Json.

```

struct SensorSetting InitJsonObject(struct SensorSetting msg)
{
    msg.ClientID = mqtt_clientID;
    msg.TimeZone = TIMEZONE;
    msg.Interval = String(SendPeriod);
    msg.Properties = PROP;
    return msg;
}

```

Gambar Potongan Source Code Fungsi InitJsonObject

Dapat dilihat dari gambar diatas, fungsi InitJsonObject memiliki input objek SensorSetting dan output objek SensorSetting. Pada fungsi ini dilakukan deklarasi awal beberapa variabel dari objek tersebut. Yang dideklarasikan antara lain zona waktu yang digunakan, interval pengiriman, dan ClientID.

Untuk melakukan pembacaan data accelerometer dan magnetometer dari MPU9255 diperlukan beberapa fungsi dasar yaitu, Filter, I2Cread dan I2CwriteByte. Fungsi Filter untuk mengimplementasikan filter lowpass terhadap data accelerometer, fungsi I2Cread untuk membaca data dr suatu device, dan I2CwriteByte digunakan untuk menuliskan data 8 bit thdp suatu device I2C.

```

double Filter(double In, double OutLPS, int FZK)
{
    if(FZK > 0)
    {
        double schritt1 = OutLPS - In;
        double schritt2 = 1.0/FZK;
        double schritt3 = schritt1 * schritt2;
        double wert = OutLPS - schritt3;

        return wert;
    }
    else
    {
        return (In);
    }
}

// This function read Nbytes bytes from I2C device at address Address.
// Put read bytes starting at register Register in the Data array.
void I2Cread(uint8_t Address, uint8_t Register, uint8_t Nbytes, uint8_t* Data)
{
    // Set register address
    Wire.beginTransmission(Address);
    Wire.write(Register);
    Wire.endTransmission();

    // Read Nbytes
    Wire.requestFrom(Address, Nbytes);
    uint8_t index=0;
    while (Wire.available())
        Data[index++]=Wire.read();
}

// Write a byte (Data) in device (Address) at register (Register)
void I2CwriteByte(uint8_t Address, uint8_t Register, uint8_t Data)
{
    // Set register address
    Wire.beginTransmission(Address);
    Wire.write(Register);
    Wire.write(Data);
    Wire.endTransmission();
}

void MPU9255_Init()
{
    Wire.begin(SDA_PIN,SCL_PIN);

    // Set accelerometers low pass filter at 5Hz
    I2CwriteByte(MPU9250_ADDRESS,29,0x06);
}

```

```

// Configure accelerometers range
I2CwriteByte(MPU9250_ADDRESS,28,ACC_FULL_SCALE_2_G);
// accelerometers low pass filter off
I2CwriteByte(MPU9250_ADDRESS,29,ACC_FILTER_OFF);
// Set by pass mode for the magnetometers
I2CwriteByte(MPU9250_ADDRESS,0x37,0x02);

// Request continuous magnetometer measurements in 16 bits
I2CwriteByte(MAG_ADDRESS,0x0A,0x16);

}

```

Dapat dilihat pada fungsi MPU9255_Init(), hal pertama yang dilakukan adalah menginisiasi penggunaan port untuk I2C. Setelah itu dilakukan untuk mengeset low pass filter 5 Hz, mengatur range pembacaan data accelerometer menjadi antara -2G sampai 2G, mematikan fitur lowpass filter pada MPU9255, mengatur magnetometer menjadi mode pass, dan mengatur untuk pembacaan magnetometer secara kontinu dalam 16 bits.

```

void WiFiConnect()
{
    // We start by connecting to a WiFi network
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    randomSeed(micros());
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

```

Gambar Potongan Source Code Fungsi WiFiConnect

Pada fungsi WiFiConnect dilakukan proses inisiasi koneksi antara NodeMCU dengan Access Point. Fungsi ini akan terus menerus berjalan sampai ada Access Point yang terhubung. Status dari proses ini dapat dilihat pada serial monitor pada IDE Arduino.

```

void loop()
{
    if(!client.loop()) client.connect(mqtt_clientID.c_str(), mqtt_user.c_str(), mqtt_password.c_str());

    if (!client.connected())
    {
        reconnect_server();
        i = 0;
    }
    else
    {
        t.now = millis();
        data = Acc_Read();
        t imu = millis() - t.now;
        yield();
        payload_data.acc[i].x = char_repr(data.x);
        payload_data.acc[i].y = char_repr(data.y);
        payload_data.acc[i].z = char_repr(data.z);
        if(i==0)
        {
            String YEAR = String(gps.date.year());
            String MONTH = String(gps.date.month());
            String DATE = String(gps.date.day());
            String HOUR = String(gps.time.hour());
            String MINUTE = String(gps.time.minute());
            String SECOND = String(gps.time.second());
            payload_data.PointTime = YEAR + "-" + MONTH + "-" + DATE + "T" + HOUR + ":" + MINUTE + ":" + SECOND + "Z";
        }
        i++;
        if(i==N)
        {
            i = 0;
            String message = JsonToString(payload_data,payload_setting);
            char message_t[MQTT_MAX_PACKET_SIZE];
            message.toCharArray(message_t, MQTT_MAX_PACKET_SIZE);

            t.now = millis();
            bool test = client.publish(server_topic, message_t);
            t.mqtt = millis()-t.now;

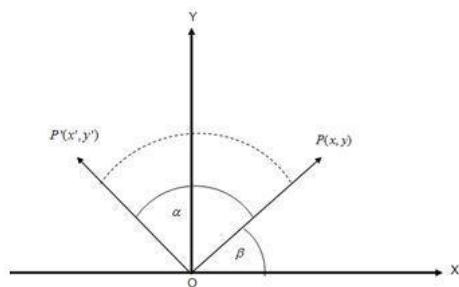
            if(test){
                Serial.print("publish success ");
                Serial.print(String(t.imu));
                Serial.print(" ");
                Serial.println(String(t.mqtt));
            }
        }
    }
    delay((1000/NData)-t imu);
}

```

Gambar Potongan Source Code Loop

Gambar diatas merupakan implementasi algoritma ECN pada saat loop(). Algoritma akan memeriksa apakah sensor terkoneksi dengan server. Jika tidak maka sensor akan terus berusaha untuk melakukan koneksi ulang ke server sampai berhasil. Saat berhasil maka sensor akan mulai untuk melakukan sampling sebanyak N. Pada saat sampling pertama, sensor akan melakukan pembacaan timestamp dan disimpan pada objek Json. Hasil sampling akan diubah representasinya dari float menjadi char. Hal ini bertujuan untuk mengurangi lebar message yang akan dikirim ke server. Ketika sampling selesai dilakukan, Objek Json akan diubah menjadi string dengan format json. String tersebut diubah menjadi char array yang akan dikirimkan ke server. Publish message ke server dilakukan dengan fungsi publish. Setelah itu akan dilakukan fungsi delay yang lamanya tergantung dari lama proses sampling dalam pembacaan sensor IMU MPU9255.

Pada fungsi Acc_Read dilakukan beberapa proses pembacaan, yaitu pembacaan data accelerometer, temperatur, dan magnetometer. Data temperatur diperlukan untuk memonitor kondisi sensor, krn sensitivitas sensor akan terganggu ketika suhu naik ke tingkat yang seharusnya. Data magnetometer diperlukan untuk menyesuaikan pembacaan data accelerometer terhadap peletakan sensor. Idealnya sumbu x IMU harus searah dengan kutub utara, ketika terjadi penyimpangan maka pembacaan data accelerometer menjadi berbeda antara beberapa sensor. Sehingga pembacaan accelerometer yang seharusnya memenuhi persamaan sebagai berikut.



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Dengan nilai x' dan y' merupakan nilai percepatan dalam sumbu x dan y dengan x searah dengan kutub utara, x dan y adalah nilai percepatan dalam sumbu x dan y yang belum disesuaikan, dan α merupakan simpangan antara sumbu x IMU dengan kutub utara. Untuk metode pembacaan accelerometer dan magnetometer dapat dilakukan dengan membaca dari register 0x3B dan 8 address setelahnya untuk accelerometer dan dari register 0x03 dan 7 address setelahnya untuk magnetometer. Tetapi setelah melakukan pembacaan accelerometer, register data ready harus ditunggu terlebih dahulu sampai bernilai 1, setelah itu pembacaan data magnetometer dapat dilakukan. Untuk lebih jelasnya dapat dilihat source code lengkapnya pada lampiran.

```

void reconnect_server() {
    // Loop until we're reconnected
    while (!client.connected())
    {
        // Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        //if you MQTT broker has clientID,username and password
        //please change following line to      if (client.connect(clientId,userName,password))
        if (client.connect(mqtt_clientID.c_str(), mqtt_user.c_str(), mqtt_password.c_str()))
        {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.print(" try again in ");
            Serial.print(String(SendPeriod));
            // Wait 1 Sending Period before retrying
            delay(SendPeriod);
        }
    }
} //end reconnect()

```

Gambar Potongan Source Code Fungsi reconnect_server

Pada fungsi reconnect_server dilakukan proses inisiasi koneksi ulang lagi ke messaging server CloudAMQP / Server PPTIK. Proses ini terus berlangsung sampai NodeMCU dapat terkoneksi ke server lagi. Status dari proses reconnect server ini dapat dilihat pada serial monitor. Penyebab error dalam percobaan koneksi ke server dapat dilihat dari nilai rc-nya. Informasi mengenai alasan error-nya sangat berguna untuk melakukan proses debugging. Keterangan mengenai nilai rc dapat dilihat pada gambar berikut.

- -4 : MQTT_CONNECTION_TIMEOUT - the server didn't respond within the keepalive time
- -3 : MQTT_CONNECTION_LOST - the network connection was broken
- -2 : MQTT_CONNECT_FAILED - the network connection failed
- -1 : MQTT_DISCONNECTED - the client is disconnected cleanly
- 0 : MQTT_CONNECTED - the client is connected
- 1 : MQTT_CONNECT_BAD_PROTOCOL - the server doesn't support the requested version of MQTT
- 2 : MQTT_CONNECT_BAD_CLIENT_ID - the server rejected the client identifier
- 3 : MQTT_CONNECT_UNAVAILABLE - the server was unable to accept the connection
- 4 : MQTT_CONNECT_BAD_CREDENTIALS - the username/password were rejected
- 5 : MQTT_CONNECT_UNAUTHORIZED - the client was not authorized to connect

Gambar Informasi Mengenai Nilai RC dan Penjelasannya

```

String JsonToString(struct MessageData msg, struct SensorSetting set)
{
    String a = "";
    a = a + "{" + " \\"pointTime\":" + " + \"\"" + msg.PointTime + "\"\"" + ",";
    a = a + "\\"timeZone\":" + " + \"\"" + set.TimeZone + "\"\"" + ",";
    a = a + "\\"interval\":" + " + \"\"" + set.Interval + "\"\"" + ",";
    a = a + "\\"clientID\":" + " + \"\"" + set.ClientID + "\"\"" + ",";
    a = a + "\\"geojson\":" + " + "{";
    //a = a + "\\"type\":" + " + \"\"" + msg.geometry.type + "\"\"" + ",";
    a = a + "\\"geometry\":" + " + "{";
    a = a + "\\"type\":" + " + \"\"" + TYPE + "\"\"" + ",";
    a = a + "\\"coordinates\":" + " + msg.coordinates[0] + ",";
    a = a + msg.coordinates[1] + "]";
    a = a + "),";
    a = a + "\\"properties\":" + "{";
    a = a + "\\"name\":" + " + \"\"" + set.Properties + "\"\"";
    a = a + "}";
    a = a + "},";
    a = a + "\\"accelerations\":" + "[";
    for (int i = 0; i < N; i++)
    {
        if (i != (N-1))
        {
            a = a + "{";
            a = a + "\\"x\":" + msg.acc[i].x + ",";
            a = a + "\\"y\":" + msg.acc[i].y + ",";
            a = a + "\\"z\":" + msg.acc[i].z ;
            a = a + "},";
        }
        else
        {
            a = a + "{";
            a = a + "\\"x\":" + msg.acc[i].x + ",";
            a = a + "\\"y\":" + msg.acc[i].y + ",";
            a = a + "\\"z\":" + msg.acc[i].z ;
            a = a + "}";
        }
    }
    a = a + "]";
    a = a + "]";
    //Serial.println(a);
    return a;
}

```

Gambar Potongan Source Code Fungsi JsonToString

Fungsi JsonToString memiliki input objek Json MessageData dan SensorSetting yang sudah dideklarasikan sebelumnya dengan output dengan tipe data string. Tidak digunakan library untuk encode dan decode string format Json yang tersedia dikarenakan tingkat kerumitan format Json yang kita gunakan tidak dapat di-handle oleh fungsi-fungsi tersebut, oleh karena itu dibuat fungsi encode format Json ini. Langkah yang dilakukan sangat sederhana, yaitu hanya menyusun string sesuai dengan format Json sehingga terbentuk string yang bersesuaian dengan objek Json yang kita gunakan. Dilakukan proses penambahan string a sesuai dengan format Json yang diinginkan. Perhatikan bahwa “\\” digunakan untuk mencetak string petik awal atau akhir.

```

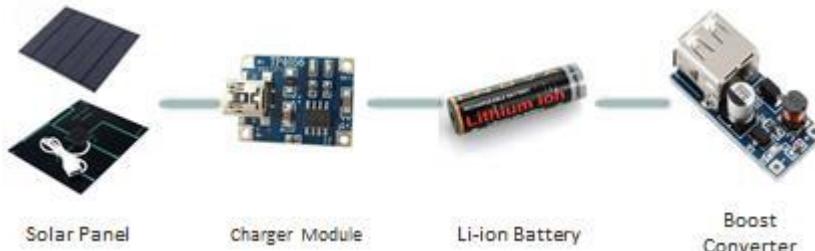
String char_repr(float x)
{
    uint8_t *temp;
    char z[2];
    int16_t y = (int16_t) (x*10000.00);
    temp = (uint8_t *) &y;
    z[0] = (char) temp[0];
    z[1] = (char) temp[1];
    String t = String(z[1]) + String(z[0]);

    return t;
}

```

Untuk mengurangi panjang message saat publish ke messaging server, digunakan teknik penggantian representasi float menjadi representasi char. Input float dikalikan dengan 10000 dan diubah tipe datanya menjadi integer 2 byte. Setelah itu pointer temp menunjuk ke alamat variabel integer 2 byte tsb. Karena setiap address pada arduino berisi 8 byte, maka 8 byte MSB dan 8 byte LSB dapat diakses dengan menggunakan pointer ini. Tiap 8 byte diubah representasinya menjadi char. Lalu kedua buah variabel char itu disatukan. Dengan teknik representasi char ini, panjang data yang di-publish ke server bisa berkurang sebesar 60%-70%.

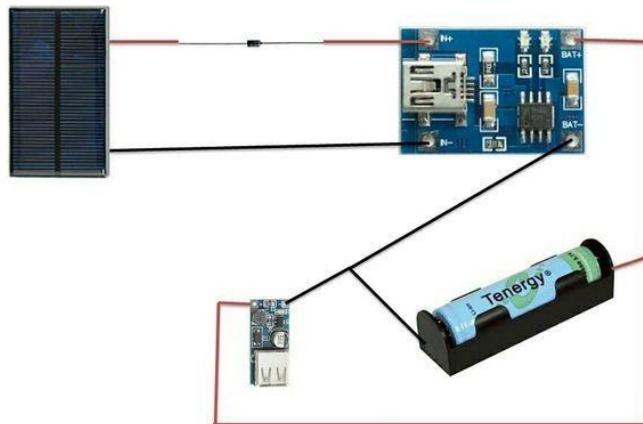
2.1.1 Charge Controller Module



Gambar Diagram Blok Charge Controller Module

Untuk membuat sensor ECN dapat berjalan secara terus menerus diperlukannya sumber daya yang dapat diperbaharui untuk menyuplai sensor ECN. Digunakan solar panel karena mudah untuk mengaksesnya serta cukup sederhana dalam proses implementasinya serta harga yang tidak terlalu mahal.

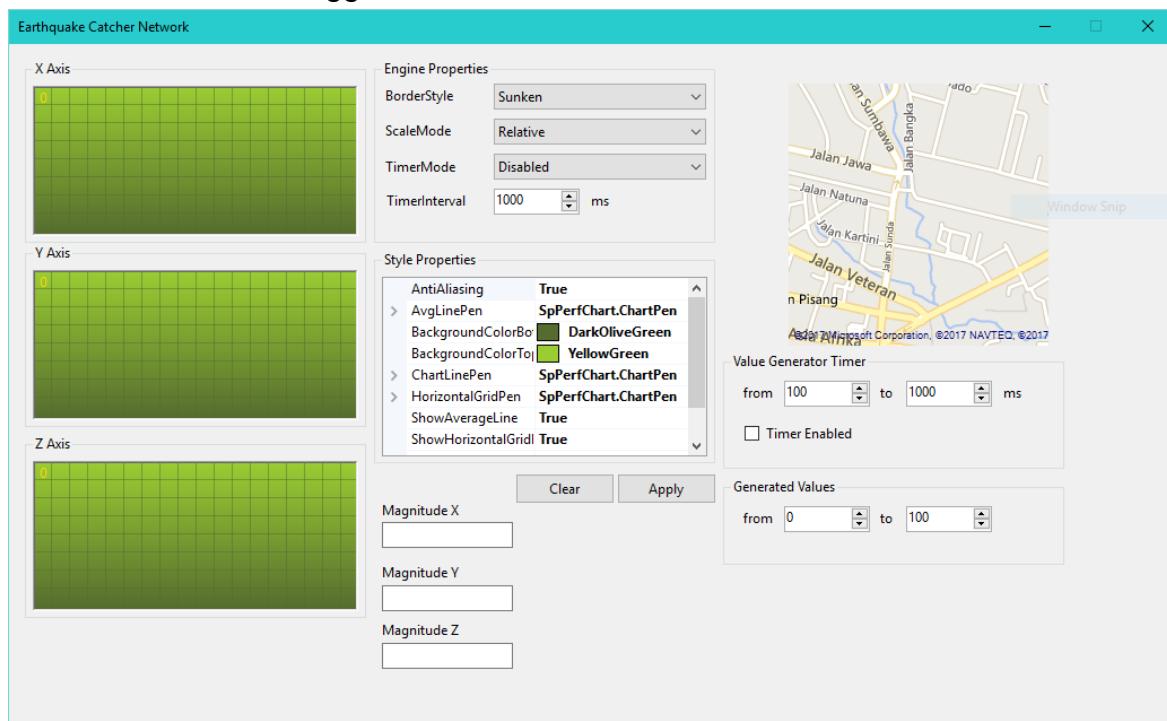
Digunakan solar panel dengan keluaran 5 Volt dan daya 4W sehingga solar panel memiliki kemampuan untuk menyuplai arus yang diperlukan NodeMCU (0,2A) dan charging baterai (0,6A) agar baterai dapat menyuplai sensor pada malam hari. Baterai yang digunakan untuk menyimpan daya adalah 2 baterai Li-Ion dengan kapasitas masing-masing adalah 2200 mAh. Pemilihan Li-Ion sendiri karena lebih tidak mudah rusaknya baterai ketika digunakan jika dibandingkan dengan baterai-baterai yang dapat di-charge lainnya seperti NiMH dan NiCD. Untuk mengatur tingkat charging baterai, digunakan modul Li-Ion battery charge module TP4056. Agar stabilnya tegangan yang diberikan oleh charge module ini, diperlukan modul boost converter. Digunakan modul boost converter dengan range input dari 0,9V – 5V dan arus keluaran antara 0,2 – 0,3 A. Untuk koneksi antar modulnya dapat dilihat pada gambar dibawah ini.



Gambar Skematik Modul Charge Controller

2.3 Perangkat Lunak Pengolah Data dan Antarmuka Grafis

2.3.1 Antarmuka Pengguna Grafis



Tampilan ini adalah tampilan awal dari antarmuka pengguna grafis sistem detektor gempa ini. Hanya data dari satu sensor yang dimasukkan ke dalam proses yang ada dibalik tampilan ini. Data tersebut berupa percepatan ke arah x,y,dan z, serta data posisi yang ditampilkan melalui peta di pojok kanan atas. Sensor akan ditandai dengan pin berwarna biru di peta tersebut. Bila ada lebih dari satu sensor yang terhubung, maka akan ada

banyak pin biru di peta. Tombol dan bar lainnya digunakan untuk melakukan pengaturan terhadap grafik data percepatan yang ditampilkan pada kotak hijau di bagian kiri. Selain itu, magnitudo dari getaran pada masing-masing sumbu juga ditampilkan secara kuantitatif di bagian bawah.

Peta tersebut disediakan oleh Bing secara gratis, melalui proyek gMap.Net, yaitu paket peta dan gps yang khusus dibuat untuk platform .NET, termasuk didalamnya C#. Program akan menerima data lokasi dari masing-masing sensor berupa lintang dan bujur, dan program lalu akan membuat sebuah pin biru di peta yang lokasinya sesuai dengan lintang dan bujur yang diterima. Pin dapat dibuat pada peta dengan jumlah yang tidak terbatas. Pengguna juga dapat menggeser peta sehingga bagian lain dari bumi yang terlihat pada layar, dan juga melakukan pembesaran atau pengecilan.

Tampilan grafik yang merepresentasikan data percepatan yang dikirim sensor bisa diatur sebagai berikut. Tiga pilihan yang terletak pada bagian atas berfungsi untuk mengatur *engine* dari grafik. Sebagai contoh, ada yang mengatur kecepatan grafik bergeser ke kiri, ada yang mengatur pembesaran/pengecilan skala grafik secara otomatis, dan juga mengatur jenis dari perbatasan grafik, apakah absolut atau bergerak mengikuti nilai grafik yang sekarang.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using RabbitMQ.Client;
using RabbitMQ.Client.Events;
using GMap.NET;

using GMap.NET.WindowsForms;
using GMap.NET.WindowsForms.Markers;
using GMap.NET.MapProviders;
using Newtonsoft.Json;
using System.IO;

namespace SimplePerfChart
{

    public partial class FrmTestingForm : Form
    {
        private object valueGenSync = new object();
        private Random randGen = new Random();
        //public static decimal data = 0;
        private int valueGenFrom = -5;
        private int valueGenTo = 5;
        private int valueGenTimerFrom = 100;
        private int valueGenTimerTo = 1000;
        public static string data = "";

        public struct accel
        {
            public decimal accel_x;
            public decimal accel_y;
            public decimal accel_z;
        }
    }
}
```

```

        }

    public FrmTestingForm()
    {
        InitializeComponent();

        this.Font = SystemInformation.MenuFont;

        propGrid.SelectedObject = perfChart.PerfChartStyle;

        // Apply default Properties
        perfChart.TimerInterval = 1000;

        // Populate DropDownList Boxes
        foreach (String item in System.Enum.GetNames(typeof(Border3DStyle)))
        {
            cmbBxBorder.Items.Add(item);
        }
        foreach (String item in
System.Enum.GetNames(typeof(SpPerfChart.ScaleMode)))
        {
            cmbBxScaleMode.Items.Add(item);
        }
        foreach (String item in
System.Enum.GetNames(typeof(SpPerfChart.TimerMode)))
        {
            cmbBxTimerMode.Items.Add(item);
        }

        // Select default values
        cmbBxTimerMode.SelectedItem = perfChart.TimerMode.ToString();
        cmbBxScaleMode.SelectedItem = perfChart.ScaleMode.ToString();
        cmbBxBorder.SelectedItem = perfChart.BorderStyle.ToString();

    }

private void chkBxTimerEnabled_CheckedChanged(object sender, EventArgs e)
{
    if (chkBxTimerEnabled.Checked && !bgWrkTimer.IsBusy)
    {
        RunTimer();
    }
}

private void RunTimer()
{
    int waitFor = randGen.Next(valueGenTimerFrom, valueGenTimerTo);
    bgWrkTimer.RunWorkerAsync(waitFor);
}

private void bgWrkTimer_DoWork(object sender, DoWorkEventArgs e)
{
    Thread.Sleep(Convert.ToInt32(e.Argument));
}

public class Config
{
    public string host;
}

```

```

        public string user;
        public string vhost;
        //public string port;
        public string password;

    }

    private static void consume_data()
    {
        //accel accel_data;
        //AccelerationReport accelerations = new AccelerationReport();
        //int genValue = randGen.Next(valueGenFrom, valueGenTo);

    }

    private void bgWrkTimer_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
    //consume_data();
    ConnectionFactory factory;
    using (StreamReader r = new StreamReader("config1.json"))
    {
        string json = r.ReadToEnd();
        Config config = JsonConvert.DeserializeObject<Config>(json);

        factory = new ConnectionFactory();// { HostName = config.host,
UserName = config.user, VirtualHost = config.vhost, Password = config.password };
        //factory.Uri =
"amqp://lsowqccg:kbLv9YbzjQwxz20NH7Rfy98TTV2eK17j@black-
boar.rmq.cloudamqp.com/lsowqccg";
        factory.Uri =
"amqp://lsowqccg:kbLv9YbzjQwxz20NH7Rfy98TTV2eK17j@black-
boar.rmq.cloudamqp.com/lsowqccg";
    }

    factory.Protocol = Protocols.DefaultProtocol;
    factory.Port = AmqpTcpEndpoint.UseDefaultPort;
    using (var connection = factory.CreateConnection())
    using (var channel = connection.CreateModel())
    {
        channel.QueueDeclare(queue: "ecn", //"emergency_gui",
                             durable: true,
                             exclusive: false,
                             autoDelete: false,
                             arguments: null);
        channel.QueueBind(queue: "ecn", //"emergency_gui",
                          exchange: "amq.topic",
                          routingKey: "amq.topic.ecn" //emergency"
                        );

        Console.WriteLine("Queue Declare Emergency GUI");
        var consumer = new EventingBasicConsumer(channel);
        consumer.Received += (model, ea) =>
        {
            var body = ea.Body;
            var message = Encoding.UTF8.GetString(body);
            Console.WriteLine(" [x] Received {0}", message);
            Console.WriteLine("/////////");
            ///////////");
            data = message;
        }
    }
}

```

```

//accel_data = ParsingMessage(data);
try
{
    //AccelerationReport accelReport =
JsonConvert.DeserializeObject<AccelerationReport>(message);
    data accelReport =
JsonConvert.DeserializeObject<data>(message);
        //Console.WriteLine("{0} {1} {2}", accel_data.accel_x,
accel_data.accel_y, accel_data.accel_z);
        //console.WriteLine("{0} {1} {2} {3}",
accelreport.geometry.coordinates, accelreport.accelerations[0].x,
accelreport.accelerations[0].y, accelreport.accelerations[0].z);

//perfchart.addValue((decimal)accelreport.accelerations[0].x * 1000);
//perfchart1.addValue((decimal)accelreport.accelerations[0].y * 1000);

//perfchart2.addValue((decimal)accelreport.accelerations[0].z * 1000);
//Console.WriteLine("{0} {1} {2} {3} {4}",
accelReport.geometryData.coordinate.lon,
accelReport.geometryData.coordinate.lng, accelReport.accelerations.x,
accelReport.accelerations.y, accelReport.accelerations.z);
    Console.WriteLine("{0} {1} {2} {3} {4}",
accelReport.geojson.geometry.coordinates[0],
accelReport.geojson.geometry.coordinates[1], accelReport.accelerations[0].x,
accelReport.accelerations[0].y, accelReport.accelerations[0].z);
        for (int i = 0 ; i < 20; i++)
    {

perfChart.AddValue((decimal)int.Parse(accelReport.accelerations[i].x));
perfChart1.AddValue((decimal)int.Parse(accelReport.accelerations[i].y));
perfChart2.AddValue((decimal)int.Parse(accelReport.accelerations[i].z));
    }
    gMapControl1.Position = new
GMap.NET.PointLatLng(double.Parse(accelReport.geojson.geometry.coordinates[0]),
double.Parse(accelReport.geojson.geometry.coordinates[1]));

}
catch (Exception ex)
{
    Console.WriteLine("ERROR: {0}", ex);
}

};

channel.BasicConsume(queue: "ecn", //"emergency_gui",
noAck: true,
consumer: consumer);

Console.WriteLine("Already BasicConsume");
//Console.ReadLine();
}

if (chkBxTimerEnabled.Checked)
{

RunTimer();
}
}

private void cmbBxBorder_SelectedIndexChanged(object sender, EventArgs e)

```

```

    {
        perfChart.BorderStyle = (Border3DStyle)Enum.Parse(
            typeof(Border3DStyle), cmbBxBorder.SelectedItem.ToString()
        );
    }

    private void cmbBxScaleMode_SelectedIndexChanged(object sender, EventArgs e)
    {
        perfChart.ScaleMode = (SpPerfChart.ScaleMode)Enum.Parse(
            typeof(SpPerfChart.ScaleMode),
        cmbBxScaleMode.SelectedItem.ToString()
        );
    }

    private void cmbBxTimerMode_SelectedIndexChanged(object sender, EventArgs e)
    {
        perfChart.TimerMode = (SpPerfChart.TimerMode)Enum.Parse(
            typeof(SpPerfChart.TimerMode),
        cmbBxTimerMode.SelectedItem.ToString()
        );
    }

    private void numUpDnTimerInterval_ValueChanged(object sender, EventArgs e)
    {
        perfChart.TimerInterval = Convert.ToInt32(numUpDnTimerInterval.Value);
    }

    private void btnApply_Click(object sender, EventArgs e)
    {
        valueGenFrom = Convert.ToInt32(numUpDnValFrom.Value);
        valueGenTo = Convert.ToInt32(numUpDnValTo.Value);
        if (valueGenTo < valueGenFrom)
        {
            valueGenTo = valueGenFrom;
            numUpDnValTo.Value = valueGenTo;
        }

        valueGenTimerFrom = Convert.ToInt32(numUpDnFromInterval.Value);
        valueGenTimerTo = Convert.ToInt32(numUpDnToInterval.Value);
        if (valueGenTimerTo < valueGenTimerFrom)
        {
            valueGenTimerTo = valueGenTimerFrom;
            numUpDnToInterval.Value = valueGenTimerTo;
        }
    }

    private void btnClear_Click(object sender, EventArgs e)
    {
        perfChart.Clear();
    }

    //public static void receive()
    //{
    //    var factory = new ConnectionFactory() { HostName = "localhost" };
    //    using (var connection = factory.CreateConnection())
    //    using (var channel = connection.CreateModel())
    //    {
    //        //channel.ExchangeDeclare(exchange: "data", type: "fanout");

    //        channel.QueueDeclare(queue: "emergency_gui",

```

```

        // durable: false,
        // exclusive: false,
        // autoDelete: false,
        // arguments: null);
    // channel.QueueBind(queue: "emergency_gui",
    //                    exchange: "amq.topic",
    //                    routingKey: "emergency");

    // var consumer = new EventingBasicConsumer(channel);
    // consumer.Received += (model, ea) =>
    //{
    //     var body = ea.Body;
    //     var message = Encoding.UTF8.GetString(body);
    //     Console.WriteLine(" [x] Received {0}", message);
    //     //data = Convert.ToDecimal(message);
    // };
    // channel.BasicConsume(queue: "emergency_gui",
    //                       noAck: true,
    //                       consumer: consumer);

    // Console.WriteLine(" Press [enter] to exit.");
    // Console.ReadLine();
// }

private void FrmTestingForm_Load(object sender, EventArgs e)
{
}

private void gMapControl1_Load(object sender, EventArgs e)
{
    gMapControl1.MapProvider =
    GMap.NET.MapProviders.BingMapProvider.Instance;
    GMap.NET.GMaps.Instance.Mode = GMap.NET.AccessMode.ServerOnly;
    gMapControl1.Position = new GMap.NET.PointLatLng(-6.9175, 107.6191);
    gMapControl1.ShowCenter = false;
    GMapOverlay markers = new GMapOverlay("markers");
    GMarker marker = new GMarkerGoogle(
        new PointLatLng(-6.890903, 107.610378),
        GMarkerGoogleType.blue_pushpin);
    markers.Markers.Add(marker);
    gMapControl1.Overlays.Add(markers);
}

private accel ParsingMessage(string messages)
{
    accel accel_data;
    string data_x, data_y, data_z;
    char[] delimiters = { '<', '>' };
    string[] parseMessage = messages.Split(delimiters);
    data_x = parseMessage[1];
    data_y = parseMessage[2];
    data_z = parseMessage[3];
    accel_data.accel_x = (int)Math.Ceiling(float.Parse(data_x) * 1000);
    accel_data.accel_y = (int)Math.Ceiling(float.Parse(data_y) * 1000);
    accel_data.accel_z = (int)Math.Ceiling(float.Parse(data_z) * 1000);

    return accel_data;
}

```

```

        private void perfChart_Load(object sender, EventArgs e)
    {
        //receive();
    }
}

class data
{
    public string pointTime;
    public string timeZone;
    public string interval;
    public geojson geojson;
    public acc[] accelerations;
};

class geojson
{
    public string type;
    public geometry geometry;
    public prop property;
};

class geometry
{
    public string type;
    public string[] coordinates;
};

class prop
{
    public string name;
};

class acc
{
    public string x;
    public string y;
    public string z;
};
}

```

2.3.2 Pengolahan Data

2.3.2.1 Akuisisi

Source Code untuk Akuisisi adalah sebagai berikut

```

package gov.usgs.volcanoes.swarm;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;

```

```

import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.io.File;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;

import javax.swing.BorderFactory;
import javax.swing.DefaultListModel;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JFileChooser;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.KeyStroke;
import javax.swing.event.MenuEvent;
import javax.swing.event.MenuListener;

import gov.usgs.plot.data.file.FileType;
import gov.usgs.volcanoes.core.ui.ExtensionFileFilter;
import gov.usgs.volcanoes.core.util.StringUtils;
import gov.usgs.volcanoes.swarm.data.CachedDataSource;
import gov.usgs.volcanoes.swarm.data.FileDataSource;
import gov.usgs.volcanoes.swarm.internalFrame.InternalFrameListener;
import gov.usgs.volcanoes.swarm.internalFrame.SwarmInternalFrames;
import gov.usgs.volcanoes.swarm.map.MapFrame;
import gov.usgs.volcanoes.swarm.wave.WaveClipboardFrame;

/**
 *
 * @author Dan Cervelli
 */
public class SwarmMenu extends JMenuBar implements InternalFrameListener {
    private static final long serialVersionUID = 1L;
    private JMenu fileMenu;
    private JMenuItem openFile;
    private JMenuItem closeFiles;
    private JMenuItem clearCache;
    private JMenuItem exit;

    private JMenuItem options;

```

```

private String lastLayoutName;
private JMenu layoutMenu;
private JMenuItem saveLayout;
private JMenuItem saveLastLayout;
private JMenuItem removeLayouts;

private JMenu windowMenu;
private JMenuItem tileWaves;
private JMenuItem tileHelicorders;
private JMenuItem fullScreen;
private JCheckBoxMenuItem clipboard;
private JCheckBoxMenuItem chooser;
private JCheckBoxMenuItem map;
private JMenuItem mapToFront;
private JMenuItem closeAll;

private JMenu helpMenu;
private JMenuItem about;

private AboutDialog aboutDialog;

private Map<JInternalFrame, InternalFrameMenuItem> windows;
private Map<SwarmLayout, JMenuItem> layouts;

public SwarmMenu() {
    super();
    windows = new HashMap<JInternalFrame, InternalFrameMenuItem>();
    layouts = new HashMap<SwarmLayout, JMenuItem>();
    createFileMenu();
    createLayoutMenu();
    createWindowMenu();
    createHelpMenu();
    SwarmInternalFrames.addInternalFrameListener(this);
}

private void createFileMenu() {
    fileMenu = new JMenu("File");
    fileMenu.setMnemonic('F');

    openFile = new JMenuItem("Open File...");
    openFile.setMnemonic('O');
    openFile.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JFileChooser chooser = FileChooser.getFileChooser();
            chooser.resetChoosableFileFilters();
            for (FileType ft : FileType.getKnownTypes()) {
                ExtensionFileFilter f = new ExtensionFileFilter(ft.extension, ft.description);
                chooser.addChoosableFileFilter(f);
            }
        }
    });
}

```

```

chooser.setFileFilter(chooser.getAcceptAllFileFilter());
File lastPath = new File(SwarmConfig.getInstance().lastPath);
chooser.setCurrentDirectory(lastPath);
chooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
chooser.setMultiSelectionEnabled(true);
chooser.setDialogTitle("Open Wave as Data Source");
int result = chooser.showOpenDialog(Swarm.getApplicationFrame());
if (result == JFileChooser.APPROVE_OPTION) {
    File[] fs = chooser.getSelectedFiles();
    FileDataSource.getInstance().openFiles(fs);
}
});
openFile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
KeyEvent.CTRL_DOWN_MASK));
fileMenu.add(openFile);

closeFiles = new JMenuItem("Close Files");
closeFiles.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        FileDataSource.getInstance().flush();
    }
});
closeFiles.setMnemonic('l');
fileMenu.add(closeFiles);

clearCache = new JMenuItem("Clear Cache");
clearCache.setMnemonic('C');
clearCache.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        CachedDataSource cache = CachedDataSource.getInstance();
        if (cache != null)
            cache.flush();
    }
});
clearCache.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F12,
KeyEvent.CTRL_DOWN_MASK));
fileMenu.add(clearCache);
fileMenu.addSeparator();

options = new JMenuItem("Options...");
options.setMnemonic('O');
options.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OptionsDialog od = new OptionsDialog();
        od.setVisible(true);
    }
});
fileMenu.add(options);

```

```

fileMenu.addSeparator();

exit = new JMenuItem("Exit");
exit.setMnemonic('x');
exit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Swarm.getApplication().closeApp();
    }
});
fileMenu.add(exit);
add(fileMenu);

fileMenu.addMenuListener(new MenuListener() {
    public void menuSelected(MenuEvent e) {
        CachedDataSource cache = CachedDataSource.getInstance();
        clearCache.setEnabled(!cache.isEmpty());
    }

    public void menuDeselected(MenuEvent e) {
    }

    public void menuCanceled(MenuEvent e) {
    }
});

private void createLayoutMenu() {
    layoutMenu = new JMenu("Layout");
    layoutMenu.setMnemonic('L');

    saveLayout = new JMenuItem("Save Layout...");
    saveLayout.setMnemonic('S');
    saveLayout.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Swarm.getApplication().saveLayout(null);
        }
    });
    saveLayout.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_L,
KeyEvent.CTRL_DOWN_MASK));
    layoutMenu.add(saveLayout);

    saveLastLayout = new JMenuItem("Overwrite Last Layout...");
    saveLastLayout.setMnemonic('L');
    saveLastLayout.setEnabled(false);
    saveLastLayout.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if (lastLayoutName != null)
                Swarm.getApplication().saveLayout(lastLayoutName);
        }
    });
}

```

```

        saveLastLayout.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_L,
KeyEvent.CTRL_DOWN_MASK
        | KeyEvent.SHIFT_DOWN_MASK));
layoutMenu.add(saveLastLayout);

removeLayouts = new JMenuItem("Remove Layout...");  

removeLayouts.setMnemonic('R');
removeLayouts.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        RemoveLayoutDialog d = new RemoveLayoutDialog();
        d.setVisible(true);
    }
});
layoutMenu.add(removeLayouts);
layoutMenu.addSeparator();
add(layoutMenu);
}

public String getLastLayoutName() {
    return lastLayoutName;
}

public void setLastLayoutName(String ln) {
    lastLayoutName = ln;
    saveLastLayout.setEnabled(true);
    saveLastLayout.setText("Overwrite Last Layout (" + ln + ")");
}

public void addLayout(final SwarmLayout sl) {
    JMenuItem mi = new JMenuItem(sl.getName());
    mi.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            sl.process();
            setLastLayoutName(sl.getName());
        }
    });
    int i;
    for (i = 4; i < layoutMenu.getItemCount(); i++) {
        JMenuItem m = layoutMenu.getItem(i);
        if (m.getText().compareToIgnoreCase(sl.getName()) >= 0) {
            layoutMenu.add(mi, i);
            break;
        }
    }
    if (i == layoutMenu.getItemCount())
        layoutMenu.add(mi, i);
    layouts.put(sl, mi);
}

public void removeLayout(SwarmLayout sl) {

```

```

JMenuItem mi = layouts.get(sl);
layoutMenu.remove(mi);
layouts.remove(sl);
}

private void createWindowMenu() {
    windowMenu = new JMenu("Window");
    windowMenu.setMnemonic('W');

    chooser = new JCheckBoxMenuItem("Data Chooser");
    chooser.setMnemonic('D');
    chooser.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

Swarm.getApplication().setChooserVisible(!Swarm.getApplication().isChooserVisible());
    }
});
    chooser.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_D,
KeyEvent.CTRL_DOWN_MASK));
    windowMenu.add(chooser);

    clipboard = new JCheckBoxMenuItem("Wave Clipboard");
    clipboard.setMnemonic('W');
    clipboard.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            WaveClipboardFrame waveClipboard = WaveClipboardFrame.getInstance();
            waveClipboard.setVisible(!waveClipboard.isVisible());
        }
});
    clipboard.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_W,
KeyEvent.CTRL_DOWN_MASK));
    windowMenu.add(clipboard);

    final MapFrame mapFrame = MapFrame.getInstance();
    map = new JCheckBoxMenuItem("Map");
    map.setMnemonic('M');
    map.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            mapFrame.setVisible(!mapFrame.isVisible());
        }
});
    map.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_M,
KeyEvent.CTRL_DOWN_MASK));
    windowMenu.add(map);

    mapToFront = new JMenuItem("Bring Map to Front");
    mapToFront.setMnemonic('F');
    mapToFront.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            mapFrame.setVisible(true);
        }
});

```

```

        }
    });
mapToFront.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_M, 0));
windowMenu.add(mapToFront);
windowMenu.addSeparator();

tileHelicorders = new JMenuItem("Tile Helicorders");
tileHelicorders.setMnemonic('H');
tileHelicorders.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Swarm.getApplication().tileHelicorders();
    }
});
windowMenu.add(tileHelicorders);

tileWaves = new JMenuItem("Tile Waves");
tileWaves.setMnemonic('v');
tileWaves.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Swarm.getApplication().tileWaves();
    }
});
windowMenu.add(tileWaves);

windowMenu.addSeparator();

fullScreen = new JMenuItem("Kiosk Mode");
fullScreen.setMnemonic('K');
fullScreen.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Swarm.getApplication().toggleFullScreenMode();
    }
});
fullScreen.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F11, 0));
windowMenu.add(fullScreen);

windowMenu.addMenuListener(new MenuListener() {
    public void menuSelected(MenuEvent e) {
        clipboard.setSelected(WaveClipboardFrame.getInstance().isVisible());
        chooser.setSelected(Swarm.getApplication().isChooserVisible());
        map.setSelected(MapFrame.getInstance().isVisible());
    }

    public void menuDeselected(MenuEvent e) {
    }

    public void menuCanceled(MenuEvent e) {
    }
});

```

```

windowMenu.addSeparator();

closeAll = new JMenuItem("Close All");
closeAll.setMnemonic('C');
closeAll.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        SwarmInternalFrames.removeAllFrames();
    }
});
windowMenu.add(closeAll);

add(windowMenu);
}

private void createHelpMenu() {
    helpMenu = new JMenu("Help");
    helpMenu.setMnemonic('H');
    about = new JMenuItem("About...");
    about.setMnemonic('A');
    aboutDialog = new AboutDialog();
    about.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            aboutDialog.update();
            aboutDialog.setVisible(true);
        }
    });
    helpMenu.add(about);
    add(helpMenu);
}

private class InternalFrameMenuItem extends JMenuItem {
    private static final long serialVersionUID = 1L;
    private JInternalFrame frame;

    public InternalFrameMenuItem(JInternalFrame f) {
        frame = f;
        setText(f.getTitle());
        setIcon(f.getFrameIcon());
        addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    if (frame.isIcon())
                        frame.setIcon(false);
                    frame.toFront();
                    frame.setSelected(true);
                } catch (Exception ex) {
                }
            }
        });
    }
}

```

```

        }

    }

private class RemoveLayoutDialog extends SwarmModalDialog {
    private static final long serialVersionUID = 1L;
    private JList layoutList;
    private DefaultListModel model;

    protected RemoveLayoutDialog() {
        super(Swarm.getApplicationFrame(), "Remove Layouts");
        setSizeAndLocation();
    }

    protected void createUI() {
        super.createUI();
        Set<String> keys = swarmConfig.layouts.keySet();
        List<String> sls = new ArrayList<String>();
        sls.addAll(keys);
        Collections.sort(sls, String_UTILS.getCaseInsensitiveStringComparator());
        model = new DefaultListModel();
        for (String sl : sls)
            model.addElement(sl);
        layoutList = new JList(model);
        JPanel panel = new JPanel(new BorderLayout());
        panel.setBorder(BorderFactory.createEmptyBorder(5, 9, 5, 9));
        int h = Math.max(200, Math.min(350, sls.size() * 19));
        panel.setPreferredSize(new Dimension(200, h));
        panel.add(new JLabel("Select layouts to remove:"), BorderLayout.NORTH);
        panel.add(new JScrollPane(layoutList), BorderLayout.CENTER);
        mainPanel.add(panel, BorderLayout.CENTER);
    }

    public void wasOK() {
        Object[] toRemove = layoutList.getSelectedValues();

        for (Object key : toRemove) {
            SwarmLayout layout = swarmConfig.layouts.get((String)key);
            if (layout != null) {
                JMenuItem mi = layouts.get(layout);
                layoutMenu.remove(mi);
                swarmConfig.removeLayout(layout);
            }
        }
    }

    public void internalFrameAdded(JInternalFrame f) {
        InternalFrameMenuItem mi = new InternalFrameMenuItem(f);
        windows.put(f, mi);
        windowMenu.add(mi);
    }
}

```

```

    }

    public void internalFrameRemoved(JInternalFrame f) {
        InternalFrameMenuItem mi = windows.get(f);
        windows.remove(f);
        windowMenu.remove(mi);
    }
}

```

2.3.2.2 Pemrosesan

2.3.2.2.1 Master

SCmaster dirancang sebagai semacam microkernel atau mediator yang delegasi permintaan klien. Oleh karena itu adalah aplikasi utama yang bertanggung jawab untuk orkestrasi dari sistem terdistribusi. Untuk berpartisipasi dalam sistem terdistribusi klien perlu mengirim permintaan connect ke scmaster tersebut. Pada gilirannya master mengembalikan pesan pengakuan yang baik menginformasikan klien masuk atau penolakan. Jika permintaan connect berhasil pesan pengakuan akan memberikan klien dengan kelompok pesan yang tersedia dapat berlangganan. Selain itu, semua klien yang terkoneksi akan diberitahu tentang anggota baru bergabung. Dalam kasus master dikonfigurasi dengan database klien juga akan menerima langsung menindaklanjuti pesan yang memegang alamat database ini. Alamat dapat digunakan untuk mengambil data arsip nanti. Setelah sambungan sudah didirikan setiap pesan akan melewati master pertama di mana itu diproses sesuai dan kemudian diteruskan ke kelompok sasaran. Setelah klien dilakukan dengan memproses pesan putuskan akan dikirim ke master yang pada gilirannya memberitahu semua klien yang tersisa tentang meninggalkannya.

```

package gov.usgs.volcanoes.swarm;

import com.jgoodies.looks.plastic.Plastic3DLookAndFeel;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.awt.Dimension;
import java.awt.Frame;
import java.awt.KeyboardFocusManager;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.util.ArrayList;

import javax.swing.AbstractAction;

```

```

import javax.swing.BorderFactory;
import javax.swing.JDesktopPane;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.KeyStroke;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

import gov.usgs.plot.data.Wave;
import gov.usgs.volcanoes.core.configfile.ConfigFile;
import gov.usgs.volcanoes.core.quakeml.Event;
import gov.usgs.volcanoes.core.time.CurrentTime;
import gov.usgs.volcanoes.core.time.J2kSec;
import gov.usgs.volcanoes.core.ui.GlobalKeyManager;
import gov.usgs.volcanoes.core.util.StringUtils;
import gov.usgs.volcanoes.swarm.chooser.DataChooser;
import gov.usgs.volcanoes.swarm.data.CachedDataSource;
import gov.usgs.volcanoes.swarm.data.SeismicDataSource;
import gov.usgs.volcanoes.swarm.event.EventFrame;
import gov.usgs.volcanoes.swarm.heli.HelicorderViewerFrame;
import gov.usgs.volcanoes.swarm.internalFrame.InternalFrameListener;
import gov.usgs.volcanoes.swarm.internalFrame.SwarmInternalFrames;
import gov.usgs.volcanoes.swarm.map.MapFrame;
import gov.usgs.volcanoes.swarm.rsam.RsamViewerFrame;
import gov.usgs.volcanoes.swarm.wave.MultiMonitor;
import gov.usgs.volcanoes.swarm.wave.WaveClipboardFrame;
import gov.usgs.volcanoes.swarm.wave.WaveViewPanel;
import gov.usgs.volcanoes.swarm.wave.WaveViewerFrame;

/**
 * The main UI and application class for Swarm. Only functions directly pertaining to the
 * UI and
 * overall application operation belong here.
 *
 * TODO: resize listener TODO: chooser visibility TODO: name worker thread for better
 * debugging
 *
 * @author Dan Cervelli, Peter Cervelli, and Thomas Parker.
 */
public class Swarm extends JFrame implements InternalFrameListener {

    private static final Logger LOGGER = LoggerFactory.getLogger(Swarm.class);

    private static final long serialVersionUID = -1;
    private static Swarm application;
    private static JFrame applicationFrame;
    private static JDesktopPane desktop;
    private JPanel split;

```

```

private SwarmMenu swarmMenu;
private final CachedDataSource cache;

private static final String TITLE = "Swarm";

private static final int LEFT = 1;
private static final int RIGHT = 2;
private static final int TOP = 3;
private static final int BOTTOM = 4;
private static final int BOTTOM_LEFT = 5;
private static final int BOTTOM_RIGHT = 6;
private static final int TOP_LEFT = 7;
private static final int TOP_RIGHT = 8;

private static boolean fullScreen = false;
private int oldState = 0;
private Dimension oldSize;
private Point oldLocation;
private AbstractAction toggleFullScreenAction;
private static SwarmConfig config;
private String lastLayout = "";

public Swarm(final String[] args) {
    super(TITLE + " [" + Version.POM_VERSION + "]");
    LOGGER.info("Swarm version/date: " + Version.VERSION_STRING);
    application = this;
    applicationFrame = this;
    setIconImage(Icons.swarm.getImage());

    config = SwarmConfig.getInstance();
    config.createConfig(args);

    cache = CachedDataSource.getInstance();

    SwarmInternalFrames.addInternalFrameListener(this);
    checkJavaVersion();
    setupGlobalKeys();
    createUI();
}

private void checkJavaVersion() {
    final String version = System.getProperty("java.version");
    LOGGER.info("java.version: " + version);
    if (version.startsWith("1.1") || version.startsWith("1.2") || version.startsWith("1.3")
        || version.startsWith("1.4")) {
        JOptionPane.showMessageDialog(this, String
            .format("%s %s requires at least Java version 1.5 or above.", TITLE,
            Version.POM_VERSION),
            "Error", JOptionPane.ERROR_MESSAGE);
        System.exit(-1);
    }
}

```

```

}

final Runtime rt = Runtime.getRuntime();
LOGGER.info("maximum heap size: " + String.format("%d", rt.maxMemory()));
}

private void setupGlobalKeys() {
    // clean this up a bit and decide if I really want to use this ghkm
    // thingy
    final GlobalKeyManager m = GlobalKeyManager.getInstance();
    m.getInputMap().put(KeyStroke.getKeyStroke("F12"), "focus");
    m.getActionMap().put("focus", new AbstractAction() {
        private static final long serialVersionUID = -1;

        public void actionPerformed(ActionEvent e) {
            final KeyboardFocusManager kfm = KeyboardFocusManager.getCurrentKeyboardFocusManager();
            System.out.println("Focus check: " + "Current window: " + kfm.getFocusedWindow() + "\n\n"
                    + "Current component: " + kfm.getFocusOwner() + "\n");
        }
    });

    m.getInputMap().put(KeyStroke.getKeyStroke("alt F12"), "outputcache");
    m.getActionMap().put("outputcache", new AbstractAction() {
        private static final long serialVersionUID = -1;

        public void actionPerformed(ActionEvent e) {
            if (cache != null)
                cache.output();
        }
    });

    m.getInputMap().put(KeyStroke.getKeyStroke("ctrl L"), "savelayout");
    m.getActionMap().put("savelayout", new AbstractAction() {
        private static final long serialVersionUID = -1;

        public void actionPerformed(ActionEvent e) {
            saveLayout(null);
        }
    });

    m.getInputMap().put(KeyStroke.getKeyStroke("ctrl shift L"), "savelastlayout");
    m.getActionMap().put("savelastlayout", new AbstractAction() {
        private static final long serialVersionUID = -1;

        public void actionPerformed(ActionEvent e) {
            final String ll = swarmMenu.getLastLayoutName();
            if (ll != null)

```

```

        saveLayout(ll);
    }
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_F11,
"fullScreenToggle");
m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_F11,
InputEvent.CTRL_DOWN_MASK),
"fullScreenToggle");
m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_BACK_SLASH,
InputEvent.CTRL_DOWN_MASK),
"fullScreenToggle");
toggleFullScreenAction = new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    toggleFullScreenMode();
}
};

m.getActionMap().put("fullScreenToggle", toggleFullScreenAction);

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_RIGHT,
InputEvent.CTRL_DOWN_MASK),
"flushRight");
m.getActionMap().put("flushRight", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    flushRight();
}
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_LEFT,
InputEvent.CTRL_DOWN_MASK),
"flushLeft");
m.getActionMap().put("flushLeft", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    flushLeft();
}
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_UP,
InputEvent.CTRL_DOWN_MASK),
"flushTop");
m.getActionMap().put("flushTop", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {

```

```

        flushTop();
    }
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_DOWN,
InputEvent.CTRL_DOWN_MASK),
"flushBottom");
m.getActionMap().put("flushBottom", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    flushBottom();
}
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_PAGE_DOWN,
InputEvent.CTRL_DOWN_MASK),
"flushBottomRight");
m.getActionMap().put("flushBottomRight", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    flushBottomRight();
}
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_END,
InputEvent.CTRL_DOWN_MASK),
"flushBottomLeft");
m.getActionMap().put("flushBottomLeft", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    flushBottomLeft();
}
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_PAGE_UP,
InputEvent.CTRL_DOWN_MASK),
"flushTopRight");
m.getActionMap().put("flushTopRight", new AbstractAction() {
private static final long serialVersionUID = -1;

public void actionPerformed(final ActionEvent e) {
    flushTopRight();
}
});

m.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_HOME,
InputEvent.CTRL_DOWN_MASK),

```

```

        "flushTopLeft");
m.getActionMap().put("flushTopLeft", new AbstractAction() {
    private static final long serialVersionUID = -1;

    public void actionPerformed(final ActionEvent e) {
        flushTopLeft();
    }
});

}

@Deprecated
public static Swarm getApplication() {
    return application;
}

public static JFrame getApplicationFrame() {
    return applicationFrame;
}

private void createUI() {
    this.addWindowListener(new WindowAdapter() {
        @Override
        public void windowClosing(final WindowEvent e) {
            closeApp();
        }
    });
    this.addFocusListener(new FocusListener() {
        public void focusGained(final FocusEvent e) {
            // The main Swarm window has no need for the focus. If it gets
            // it
            // then it attempts to pass it on to the first helicorder,
            // failing
            // that it gives it to the first wave.
            if (SwarmInternalFrames.frameCount() > 0) {
                JInternalFrame jf = null;
                for (int i = 0; i < SwarmInternalFrames.frameCount(); i++) {
                    final JInternalFrame f = SwarmInternalFrames.getFrames().get(i);
                    if (f instanceof HelicorderViewerFrame) {
                        jf = f;
                        break;
                    }
                }
                if (jf == null)
                    jf = SwarmInternalFrames.getFrames().get(0);
                jf.requestFocus();
            }
        }
    });

    public void focusLost(final FocusEvent e) {}
```

```

});

desktop = new JDesktopPane();
desktop.setBorder(BorderFactory.createLineBorder(DataChooser.LINE_COLOR));
desktop.setDragMode(JDesktopPane.OUTLINE_DRAG_MODE);
// disable dragging in fullscreen mode
/*
 * desktop.setDesktopManager(new DefaultDesktopManager() { private static final long
 * serialVersionUID = -1; public void beginDraggingFrame(JComponent f) { if
 * (fullScreen) return;
 * else super.beginDraggingFrame(f); }
 *
 * public void dragFrame(JComponent f, int x, int y) { if (fullScreen) return; else
 * super.dragFrame(f, x, y); } });
 */
this.setSize(config.windowWidth, config.windowHeight);
this.setLocation(config.windowX, config.windowY);
if (config.windowMaximized)
    this.setExtendedState(Frame.MAXIMIZED_BOTH);

this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

final DataChooser chooser = DataChooser.getInstance();
split = SwarmUtil.createStrippedSplitPane(JSplitPane.HORIZONTAL_SPLIT, chooser,
desktop);
split.setDividerLocation(config.chooserDividerLocation);
split.setDividerSize(4);
setChooserVisible(config.chooserVisible);
chooser.setDividerLocation(config.nearestDividerLocation);

final WaveClipboardFrame waveClipboard = WaveClipboardFrame.getInstance();
desktop.add(waveClipboard);
waveClipboard.setVisible(config.clipboardVisible);
if (Swarm.config.clipboardMaximized) {
    try {
        waveClipboard.setMaximum(true);
    } catch (final Exception e) {
    }
}

final MapFrame mapFrame = MapFrame.getInstance();
SwarmInternalFrames.add(mapFrame);
mapFrame.setVisible(config.mapVisible);
if (Swarm.config.mapMaximized) {
    try {
        mapFrame.setMaximum(true);
    } catch (final Exception e) {
    }
}

```

```

mapFrame.toFront();

swarmMenu = new SwarmMenu();
this.setJMenuBar(swarmMenu);

for (final SwarmLayout sl : config.layouts.values())
    swarmMenu.addLayout(sl);

this.setVisible(true);

final long offset = CurrentTime.getInstance().getOffset();
if (Math.abs(offset) > 10 * 60 * 1000)
    JOptionPane.showMessageDialog(this,
        "Your system clock is off by more than 10 minutes.\n"
        + "This is just for your information, Swarm will not be affected by this.",
        "System Clock", JOptionPane.INFORMATION_MESSAGE);
}

public void setChooserVisible(final boolean vis) {
    if (vis) {
        split.setRightComponent(desktop);
        split.setDividerLocation(config.chooserDividerLocation);
        setContentPane(split);
    } else {
        if (isChooserVisible())
            config.chooserDividerLocation = split.getDividerLocation();
        setContentPane(desktop);
    }
    if (SwingUtilities.isEventDispatchThread())
        validate();
}

public boolean isChooserVisible() {
    return getContentPane() == split;
}

public static boolean isFullScreenMode() {
    return fullScreen;
}

public void toggleFullScreenMode() {
    setFullScreenMode(!fullScreen);
}

public void setFullScreenMode(final boolean full) {
    if (fullScreen == full)
        return;

    requestFocus();
    fullScreen = full;
}

```

```

this.dispose();
this.setUndecorated(full);
this.setResizable(!full);

final WaveClipboardFrame waveClipboard = WaveClipboardFrame.getInstance();
waveClipboard.setVisible(!full);
waveClipboard.toBack();

// Does this do anything?
this.setVisible(true);

if (full) {
    this.setJMenuBar(null);
    config.chooserDividerLocation = split.getDividerLocation();
    oldState = this.getExtendedState();
    oldSize = this.getSize();
    oldLocation = this.getLocation();
    this.setContentPane(desktop);
    this.setVisible(true);
    this.setExtendedState(Frame.MAXIMIZED_BOTH);
    desktop.setSize(this.getSize());
    desktop.setPreferredSize(this.getSize());
} else {
    this.setJMenuBar(swarmMenu);
    this.setExtendedState(oldState);
    this.setSize(oldSize);
    this.setLocation(oldLocation);
    this.setVisible(true);
    split.setRightComponent(desktop);
    split.setDividerLocation(config.chooserDividerLocation);
    this.setContentPane(split);
}

for (final JInternalFrame frame : SwarmInternalFrames.getFrames()) {
    // Why did this check isVisible()?
    // if (frame.isVisible() && frame instanceof Kioskable) {
    if (frame instanceof Kioskable) {
        final Kioskable f = (Kioskable) frame;
        f.setKioskMode(full);
    } else {
        frame.setVisible(!full);
    }
}
validate();

tileKioskFrames();
}

public void closeApp() {
    final Point p = this.getLocation();

```

```

if (this.getExtendedState() == Frame.MAXIMIZED_BOTH)
    config.windowMaximized = true;
else {
    final Dimension d = this.getSize();
    config.windowX = p.x;
    config.windowY = p.y;
    config.windowWidth = d.width;
    config.windowHeight = d.height;
    config.windowMaximized = false;
}

final WaveClipboardFrame waveClipboard = WaveClipboardFrame.getInstance();
if (waveClipboard.isMaximum())
    config.clipboardMaximized = true;
else {
    config.clipboardX = waveClipboard.getX();
    config.clipboardY = waveClipboard.getY();
    config.clipboardWidth = waveClipboard.getWidth();
    config.clipboardHeight = waveClipboard.getHeight();
    config.clipboardMaximized = false;
}
config.clipboardVisible = WaveClipboardFrame.getInstance().isVisible();

final MapFrame mapFrame = MapFrame.getInstance();
if (mapFrame.isMaximum())
    config.mapMaximized = true;
else {
    config.mapX = mapFrame.getX();
    config.mapY = mapFrame.getY();
    config.mapWidth = mapFrame.getWidth();
    config.mapHeight = mapFrame.getHeight();
    config.mapMaximized = false;
}
config.mapVisible = mapFrame.isVisible();

config.chooserDividerLocation = split.getDividerLocation();
config.chooserVisible = isChooserVisible();

DataChooser.getInstance().updateConfig(config);

config.kiosk = Boolean.toString(fullScreen);

if (config.saveConfig) {
    final ConfigFile configFile = config.toConfigFile();
    configFile.remove("configFile");
    configFile.writeToFile(config.configFilename);
}

```

```

waveClipboard.removeWaves();
try {
    for (final JInternalFrame frame : SwarmInternalFrames.getFrames())
        frame.setClosed(true);
} catch (final Exception e) {
} // doesn't matter at this point
System.exit(0);
}

public static void loadClipboardWave(final SeismicDataSource source, final String
channel) {
    final WaveViewPanel wvp = new WaveViewPanel();
    wvp.setChannel(channel);
    wvp.setDataSource(source);
    final WaveViewPanel cwvp = WaveClipboardFrame.getInstance().getSelected();
    double st = 0;
    double et = 0;
    if (cwvp == null) {
        final double now = J2kSec.now();
        st = now - 180;
        et = now;
    } else {
        st = cwvp.getStartTime();
        et = cwvp.getEndTime();
    }
    final double fst = st;
    final double fet = et;

    final SwingWorker worker = new SwingWorker() {
        WaveClipboardFrame waveClipboard = WaveClipboardFrame.getInstance();

        @Override
        public Object construct() {
            // double now = CurrentTime.nowJ2K();
            waveClipboard.getThrobber().increment();
            final Wave sw = source.getWave(channel, fst, fet);
            wvp.setWave(sw, fst, fet);
            return null;
        }

        @Override
        public void finished() {
            waveClipboard.getThrobber().decrement();
            waveClipboard.setVisible(true);
            waveClipboard.toFront();
            try {
                waveClipboard.setSelected(true);
            } catch (final Exception e) {
            }
            waveClipboard.addWave(wvp);
        }
    };
}

```

```

        }
    };
    worker.start();
}

public static WaveViewerFrame openRealtimeWave(final SeismicDataSource source,
    final String channel) {
    final WaveViewerFrame frame = new WaveViewerFrame(source, channel);
    SwarmInternalFrames.add(frame);
    return frame;
}

public static HelicorderViewerFrame openHelicorder(final SeismicDataSource source,
    final String channel, final double time) {
    source.establish();
    final HelicorderViewerFrame frame = new HelicorderViewerFrame(source, channel,
    time);
    frame.addLinkListeners();
    SwarmInternalFrames.add(frame);
    return frame;
}

public static RsamViewerFrame openRsam(final SeismicDataSource source, final String
channel) {
    final RsamViewerFrame frame = new RsamViewerFrame(source, channel);
    SwarmInternalFrames.add(frame);
    return frame;
}

// public static PickerFrame openPicker(final WaveViewPanel insetWavePanel) {
//     PickerWavePanel p = new PickerWavePanel(insetWavePanel);
//     p.setDataSource(insetWavePanel.getDataSource().getCopy());
//     PickerFrame pickerFrame = new PickerFrame();
//     pickerFrame.setVisible(true);
//     pickerFrame.requestFocus();
//     SwarmInternalFrames.add(pickerFrame);
//     pickerFrame.setBaseWave(p);
//     return pickerFrame;
// }

// public static PickerFrame openPicker(Event event) {
//     PickerFrame pickerFrame = new PickerFrame(event);
//     pickerFrame.setVisible(true);
//     pickerFrame.requestFocus();
//     SwarmInternalFrames.add(pickerFrame);
//     return pickerFrame;
// }

public void saveLayout(String name) {

```

```

final boolean fixedName = (name != null);
final SwarmLayout sl = getCurrentLayout();
boolean done = false;
while (!done) {
    if (name == null) {
        name = (String) JOptionPane.showInputDialog(this, "Enter a name for this layout:",
            "Save Layout", JOptionPane.INFORMATION_MESSAGE, null, null, lastLayout);
        name = name.trim();
    }
    if (name != null && !"".equals(name)) {
        if (Swarm.config.layouts.containsKey(name)) {
            boolean overwrite = false;
            if (!fixedName) {
                final int opt = JOptionPane.showConfirmDialog(this,
                    "A layout by that name already exists. Overwrite?", "Warning",
                    JOptionPane.YES_NO_OPTION);
                overwrite = (opt == JOptionPane.YES_OPTION);
            } else
                overwrite = true;

            if (overwrite) {
                if (fixedName)
                    JOptionPane.showMessageDialog(this, "Layout overwritten.");
                }
                swarmMenu.removeLayout(Swarm.config.layouts.get(name));
                Swarm.config.removeLayout(Swarm.config.layouts.get(name));
            }
        }
    }

    if (!Swarm.config.layouts.containsKey(name)) {
        swarmMenu.setLastLayoutName(name);
        sl.setName(name);
        sl.save();
        swarmMenu.addLayout(sl);
        Swarm.config.addLayout(sl);
        done = true;
        lastLayout = name;
    }
} else
    done = true; // cancelled
}
}

public SwarmLayout getCurrentLayout() {
    final ConfigFile cf = new ConfigFile();
    cf.put("name", "Current Layout");
    cf.put("kiosk", Boolean.toString(isFullScreenMode()));
    cf.put("kioskX", Integer.toString(getX()));
    cf.put("kioskY", Integer.toString(getY()));
}

```

```

DataChooser.getInstance().saveLayout(cf, "chooser");

final SwarmLayout sl = new SwarmLayout(cf);
int i = 0;
for (final JInternalFrame frame : SwarmInternalFrames.getFrames()) {
    if (frame instanceof HelicorderViewerFrame) {
        final HelicorderViewerFrame hvf = (HelicorderViewerFrame) frame;
        hvf.saveLayout(cf, "helicorder-" + i++);
    } else if (frame instanceof MultiMonitor) {
        final MultiMonitor mm = (MultiMonitor) frame;
        mm.saveLayout(cf, "monitor-" + i++);
    }
}

final MapFrame mapFrame = MapFrame.getInstance();
if (mapFrame.isVisible())
    mapFrame.saveLayout(cf, "map");

return sl;
}

```

```

public void flush(final int position) {

    final Dimension ds = desktop.getSize();
    JInternalFrame bingo = null;

    final WaveClipboardFrame waveClipboard = WaveClipboardFrame.getInstance();
    if (waveClipboard.isSelected()) {
        bingo = waveClipboard;
    } else
        for (final JInternalFrame frame : SwarmInternalFrames.getFrames())
            if (frame.isSelected())
                bingo = frame;

    if (bingo != null)
        switch (position) {
            case LEFT:
                bingo.setSize(ds.width / 2, ds.height);
                bingo.setLocation(0, 0);
                break;

            case RIGHT:
                bingo.setSize(ds.width / 2, ds.height);
                bingo.setLocation(ds.width / 2, 0);
                break;

            case TOP:
                bingo.setSize(ds.width, ds.height / 2);

```

```

bingo.setLocation(0, 0);
break;

case BOTTOM:
bingo.setSize(ds.width, ds.height / 2);
bingo.setLocation(0, ds.height / 2);
break;

case BOTTOM_LEFT:
bingo.setSize(ds.width / 2, ds.height / 2);
bingo.setLocation(0, ds.height / 2);
break;

case BOTTOM_RIGHT:
bingo.setSize(ds.width / 2, ds.height / 2);
bingo.setLocation(ds.width / 2, ds.height / 2);
break;

case TOP_LEFT:
bingo.setSize(ds.width / 2, ds.height / 2);
bingo.setLocation(0, 0);
break;

case TOP_RIGHT:
bingo.setSize(ds.width / 2, ds.height / 2);
bingo.setLocation(ds.width / 2, 0);
break;
}

}

public void flushLeft() {
flush(LEFT);
}

public void flushRight() {
flush(RIGHT);
}

public void flushTop() {
flush(TOP);
}

public void flushBottom() {
flush(BOTTOM);
}

public void flushBottomRight() {
flush(BOTTOM_RIGHT);
}

```

```

public void flushBottomLeft() {
    flush(BOTTOM_LEFT);
}

public void flushTopRight() {
    flush(TOP_RIGHT);
}

public void flushTopLeft() {
    flush(TOP_LEFT);
}

public void tileKioskFrames() {
    final Dimension ds = desktop.getSize();

    final ArrayList<JInternalFrame> ks = new ArrayList<JInternalFrame>();
    for (final JInternalFrame frame : SwarmInternalFrames.getFrames()) {
        if (frame.isVisible() && frame instanceof Kioskable)
            ks.add(frame);
    }

    if (ks.size() == 0)
        return;

    int mapCount = 0;
    int heliCount = 0;
    int monitorCount = 0;
    for (final JInternalFrame frame : ks) {
        if (frame instanceof MapFrame)
            mapCount++;
        else if (frame instanceof HelicorderViewerFrame)
            heliCount++;
        else if (frame instanceof MultiMonitor)
            monitorCount++;
    }

    if (ks.size() == 4) {
        final int w = ds.width / 2;
        final int h = ds.height / 2;
        final JInternalFrame hvf0 = ks.get(0);
        final JInternalFrame hvf1 = ks.get(1);
        final JInternalFrame hvf2 = ks.get(2);
        final JInternalFrame hvf3 = ks.get(3);
        hvf0.setSize(w, h);
        hvf0.setLocation(0, 0);
        hvf1.setSize(w, h);
        hvf1.setLocation(w, 0);
        hvf2.setSize(w, h);
        hvf2.setLocation(0, h);
        hvf3.setSize(w, h);
    }
}

```

```

        hvf3.setLocation(w, h);
    } else if (ks.size() == 3 && mapCount == 1 && heliCount == 1 && monitorCount ==
1) {
        final int w = ds.width / 2;
        final int h = ds.height / 2;
        for (final JInternalFrame frame : ks) {
            if (frame instanceof MapFrame) {
                frame.setLocation(w, h);
                frame.setSize(w, h);
            } else if (frame instanceof HelicorderViewerFrame) {
                frame.setLocation(0, 0);
                frame.setSize(w, h * 2);
            } else if (frame instanceof MultiMonitor) {
                frame.setLocation(w, 0);
                frame.setSize(w, h);
            }
        }
    } else {
        final int w = ds.width / ks.size();
        int cx = 0;
        for (int i = 0; i < ks.size(); i++) {
            final JInternalFrame hvf = ks.get(i);
            try {
                hvf.setIcon(false);
                hvf.setMaximum(false);
            } catch (final Exception e) {
            }
            hvf.setSize(w, ds.height);
            hvf.setLocation(cx, 0);
            cx += w;
        }
    }
}

public void tileHelicorders() {
    final Dimension ds = desktop.getSize();

    final ArrayList<HelicorderViewerFrame> hcs = new
ArrayList<HelicorderViewerFrame>(10);
    for (final JInternalFrame frame : SwarmInternalFrames.getFrames()) {
        if (frame instanceof HelicorderViewerFrame)
            hcs.add((HelicorderViewerFrame) frame);
    }

    if (hcs.size() == 0)
        return;

    if (hcs.size() == 4) {
        final int w = ds.width / 2;
        final int h = ds.height / 2;

```

```

final HelicorderViewerFrame hvf0 = hcs.get(0);
final HelicorderViewerFrame hvf1 = hcs.get(1);
final HelicorderViewerFrame hvf2 = hcs.get(2);
final HelicorderViewerFrame hvf3 = hcs.get(3);
hvf0.setSize(w, h);
hvf0.setLocation(0, 0);
hvf1.setSize(w, h);
hvf1.setLocation(w, 0);
hvf2.setSize(w, h);
hvf2.setLocation(0, h);
hvf3.setSize(w, h);
hvf3.setLocation(w, h);
} else {
final int w = ds.width / hcs.size();
int cx = 0;
for (int i = 0; i < hcs.size(); i++) {
final HelicorderViewerFrame hvf = hcs.get(i);
try {
hvf.setIcon(false);
hvf.setMaximum(false);
} catch (final Exception e) {
}
hvf.setSize(w, ds.height);
hvf.setLocation(cx, 0);
cx += w;
}
}
}

public void tileWaves() {
final Dimension ds = desktop.getSize();

int wc = 0;
for (final JInternalFrame frame : SwarmInternalFrames.getFrames()) {
if (frame instanceof WaveViewerFrame)
wc++;
}

if (wc == 0)
return;

final int h = ds.height / wc;
int cy = 0;
for (final JInternalFrame frame : SwarmInternalFrames.getFrames()) {
if (frame instanceof WaveViewerFrame) {
final WaveViewerFrame wvf = (WaveViewerFrame) frame;
try {
wvf.setIcon(false);
wvf.setMaximum(false);
} catch (final Exception e) {
}
}
}
}

```

```

        }
        wvf.setSize(ds.width, h);
        wvf.setLocation(0, cy);
        cy += h;
    }
}
}

public static void setFrameLayer(final JInternalFrame c, final int layer) {
    desktop.setLayer(c, layer, 0);
}

private void parseKiosk() {
    final String[] kiosks = config.kiosk.split(",");
    if (config.kiosk.startsWith("layout:")) {
        final String layout = config.kiosk.substring(7);
        final SwarmLayout sl = config.layouts.get(layout);
        if (sl != null) {
            lastLayout = layout;
            sl.process();
        } else
            LOGGER.warn("could not start with layout: " + layout);
    }
    boolean set = false;
    for (int i = 0; i < kiosks.length; i++) {
        final String[] ch = kiosks[i].split(";");
        final SeismicDataSource sds = config.getSource(ch[0]);
        if (sds == null)
            continue;
        openHelicorder(sds, ch[1], Double.NaN);
        set = true;
    }
    if (config.kiosk.equals("true"))
        set = true;

    if (set)
        toggleFullScreenMode();
    else
        LOGGER.warn("no helicorders, skipping kiosk mode.");
}

public void internalFrameAdded(final JInternalFrame f) {
    desktop.add(f);
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            f.toFront();
            try {
                f.setSelected(true);
            } catch (final Exception e) {
            }
        }
    });
}

```

```

        }
    });
}

public void internalFrameRemoved(final JInternalFrame f) {
    // do nothing
}

public static void main(final String[] args) {
    try {
        UIManager.setLookAndFeel(new Plastic3DLookAndFeel());
        UIManager.put("InternalFrame.border", SwarmUtil.getInternalFrameBorder());
    } catch (final Exception e) {
    }

    final Swarm swarm = new Swarm(args);

    if (Swarm.config.isKiosk())
        swarm.parseKiosk();
}

public static EventFrame openEvent(final Event event) {
    final EventFrame eventFrame = new EventFrame(event);
    eventFrame.setVisible(true);
    eventFrame.requestFocus();
    SwarmInternalFrames.add(eventFrame);

    return eventFrame;
}
}

```

2.3.2.2 Autoloc

scautoloc adalah program SeisComP3 bertanggung jawab untuk secara otomatis mencari peristiwa seismik di dekat-real time. Ini biasanya berjalan sebagai daemon, terus membaca picks dan amplitudo dan memproses mereka secara real time. Sebuah modus offline juga tersedia. scautoloc membaca picks otomatis dan beberapa amplitudo terkait. Atas dasar itu mencoba untuk mengidentifikasi kombinasi picks yang sesuai dengan peristiwa seismik umum. Jika lokasi yang diproduksi memenuhi kriteria konsistensi tertentu, dilaporkan, yaitu diteruskan ke program lain yang mengambil asal sebagai masukan.
prosedur lokasi

Prosedur scautoloc untuk mengidentifikasi dan menemukan kejadian gempa pada dasarnya terdiri dari source code berikut:

```
package gov.usgs.volcanoes.swarm.data;
```

```

import gov.usgs.plot.data.HelicorderData;
import gov.usgs.plot.data.RSAMData;
import gov.usgs.plot.data.Wave;

import java.util.List;

import javax.swing.event.EventListenerList;

/**
 * Base class for seismic data sources.
 *
 * @author Dan Cervelli
 */
abstract public class SeismicDataSource {
    protected String name = "Unnamed Data Source";
    protected boolean storeInUserConfig = true;
    protected boolean useCache = true;
    protected int minimumRefreshInterval = 1;

    protected EventListenerList listeners = new EventListenerList();

    public Gulper createGulper(GulperList gl, String k, String ch, double t1, double t2,
int size, int delay) {
        return new Gulper(gl, k, this, ch, t1, t2, size, delay);
    }

    abstract public List<String> getChannels();

    abstract public void parse(String params);

    /**
     * Either returns the wave successfully or null if the data source could
     * not get the wave.
     *
     * @param station
     * @param t1
     * @param t2
     * @return wave if possible
     */
    abstract public Wave getWave(String station, double t1, double t2);

    abstract public HelicorderData getHelicorder(String station, double t1, double t2,
GulperListener gl);

    abstract public String toConfigString();

    protected SeismicDataSource() {
        // explicit default constructor needed for reflection
    }
}

```

```

public void addListener(SeismicDataSourceListener l) {
    listeners.add(SeismicDataSourceListener.class, l);
}

public void removeListener(SeismicDataSourceListener l) {
    listeners.remove(SeismicDataSourceListener.class, l);
}

public void fireChannelsUpdated() {
    Object[] ls = listeners.getListenerList();
    for (int i = ls.length - 2; i >= 0; i -= 2)
        if (ls[i] == SeismicDataSourceListener.class)
            ((SeismicDataSourceListener) ls[i + 1]).channelsUpdated();
}

public void fireChannelsProgress(String id, double p) {
    Object[] ls = listeners.getListenerList();
    for (int i = ls.length - 2; i >= 0; i -= 2)
        if (ls[i] == SeismicDataSourceListener.class)
            ((SeismicDataSourceListener) ls[i + 1]).channelsProgress(id,
p);
}

public void fireHelicorderProgress(String id, double p) {
    Object[] ls = listeners.getListenerList();
    for (int i = ls.length - 2; i >= 0; i -= 2)
        if (ls[i] == SeismicDataSourceListener.class)
            ((SeismicDataSourceListener) ls[i + 1]).helicorderProgress(id, p);
}

public void notifyDataNotNeeded(String station, double t1, double t2,
GulperListener gl) {
}

public void setStoreInUserConfig(boolean b) {
    storeInUserConfig = b;
}

public boolean isStoreInUserConfig() {
    return storeInUserConfig;
}

public void setUseCache(boolean b) {
    useCache = b;
}

public boolean isUseCache() {
    return useCache;
}

```

```

/**
 * Is this data source active; that is, is new data being added in real-time
 * to this data source?
 *
 * @return whether or not this is an active data source
 */
public boolean isActiveSource() {
    return false;
}

/**
 * Close the data source.
 */
public void close() {
}

public void remove() {
}

/**
 * Get a copy of the data source. The default implementation returns an
 * identical copy, that is, <code>this</code>.
 *
 * This is confusing. Why return a reference to an object the caller already has?
Should this really return a
 * clone? If so, why not call it clone()? --tjp
 *
 * @return the identical data source (this)
 */
public SeismicDataSource getCopy() {
    return this;
}

/**
 * Get a string representation of this data source. The default implementation
 * return the name of the data source.
 *
 * @return the string representation of this data source
 */
public String toString() {
    return name;
}

/**
 * Sets the data source name.
 *
 * @param s
 *          the new name
 */

```

```

public void setName(String s) {
    name = s;
}

/**
 * Gets the data source name.
 *
 * @return the name
 */
public String getName() {
    return name;
}

public void establish() {
}

public int getMinimumRefreshInterval() {
    return minimumRefreshInterval;
}

}

```

2.3.2.2.3 Amplitude

SCamp mengukur beberapa jenis amplitudo dari data gelombang. Hal mendengarkan usul dan langkah-langkah amplitudo dalam waktu jendela ditentukan dari asal. Benda-benda amplitudo yang dihasilkan dikirim ke "AMPLITUDO" kelompok messaging. banyol adalah mitra dari scmag. Biasanya, semua amplitudo dihitung sekaligus dengan banyol dan kemudian dipublikasikan. Hanya sangat jarang amplitudo perlu menghitung ulang jika lokasi asal berubah secara signifikan. amplitudo dapat digunakan kembali oleh scmag, membuat besarnya perhitungan dan pembaruan efisien. Saat ini, pemilih otomatis di SeisComP 3, scautopick, juga mengukur satu set kecil amplitudo (yaitu "SNR" dan "mb", rasio signal-to-noise dan amplitudo yang digunakan dalam mb perhitungan besarnya, masing-masing) untuk setiap pick otomatis di jendela waktu yang tetap. Jika ada sudah ada amplitudo, misalnya sebelumnya ditentukan satu per scautopick, perampok tidak akan mengukur lagi untuk aliran masing-masing. Amplitudo juga diperlukan, namun, untuk picks manual. banyol melakukan hal ini juga. Pilihan dengan berat lebih kecil dari 0,5 di sesuai Asal dibuang.

Amplitudo ini diimplementasikan dengan source code berikut:

```

package gov.usgs.volcanoes.swarm.wave;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.event.MouseEvent;
import java.awt.geom.GeneralPath;
import java.awt.geom.Line2D;

import gov.usgs.plot.Plot;
import gov.usgs.plot.data.SliceWave;

```

```

import gov.usgs.plot.data.Wave;
import gov.usgs.plot.render.TextRenderer;
import gov.usgs.plot.render.wave.SliceWaveRenderer;
import gov.usgs.volcanoes.swarm.Metadata;
import gov.usgs.volcanoes.swarm.SwarmConfig;

/**
 * A component that renders a wave in either a standard wave view, a frequency spectra,
or
 * spectrogram. Relies heavily on the Valve plotting package.
 *
 * TODO: move filter method
 *
 *
 * @author Dan Cervelli
 */
public class WaveViewPanel extends AbstractWavePanel {
    /**
     * Constructs a WaveViewPanel with default settings.
     */
    public WaveViewPanel() {
        this(new WaveViewSettings());
    }

    /**
     * Constructs a WaveViewPanel with specified settings.
     *
     * @param s the settings
     */
    public WaveViewPanel(WaveViewSettings s) {
        super(s);
    }

    /**
     * Constructs a WaveViewPanel set up the same as a source WaveViewPanel. Used when
copying a
     * waveform to the clipboard.
     *
     * @param p the source WaveViewPanel
     */
    public WaveViewPanel(WaveViewPanel p) {
        super(p);
    }

    @Override
    protected void processRightMousePress(MouseEvent e) {
        settings.cycleType();
    }

    private void paintMark(Graphics2D g2, double j2k) {

```

```

if (Double.isNaN(j2k) || j2k < startTime || j2k > endTime)
    return;

double[] t = getTranslation();
if (t == null)
    return;

double x = (j2k - t[1]) / t[0];
g2.setColor(DARK_GREEN);
g2.draw(new Line2D.Double(x, yOffset, x, getHeight() - bottomHeight - 1));

GeneralPath gp = new GeneralPath();
gp.moveTo((float) x, yOffset);
gp.lineTo((float) x - 5, yOffset - 7);
gp.lineTo((float) x + 5, yOffset - 7);
gp.closePath();
g2.setPaint(Color.GREEN);
g2.fill(gp);
g2.setColor(DARK_GREEN);
g2.draw(gp);
}

@Override
protected void annotateImage(Graphics2D g2) {
    if (!Double.isNaN(mark1))
        paintMark(g2, mark1);

    if (!Double.isNaN(mark2))
        paintMark(g2, mark2);
}

@Override
protected void processRightMouseRelease(MouseEvent e) {
    // do nothing
}

}

```

2.3.2.2.4 *Magnitude*

Menghitung besaran.

Tujuan dari scmag adalah untuk menghitung besaran. Dibutuhkan amplitudo dan asal-usul sebagai masukan dan menghasilkan StationMagnitudes dan NetworkMagnitudes sebagai output. besaran yang dihasilkan dikirim ke kelompok "BESARAN". scmag tidak mengakses bentuk gelombang apapun. Hanya menggunakan amplitudo sebelumnya dihitung, misalnya oleh perampok. Tujuan dari scmag adalah decoupling besarnya perhitungan dari pengukuran amplitudo. Hal ini memungkinkan beberapa modul untuk menghasilkan amplitudo secara bersamaan, seperti scautopick dan perampok. Begitu asal

masuk, amplitudo berkaitan dengan picks diambil baik dari buffer memori atau database untuk menghitung besaran. Saat ini jenis besaran berikut ini diimplementasikan:

```
package gov.usgs.volcanoes.swarm.heli;

import gov.usgs.volcanoes.core.configfile.ConfigFile;
import gov.usgs.volcanoes.swarm.SwarmConfig;

/**
 * Settings for a helicorder.
 *
 * TODO: eliminate this class in favor of vdx.HelicorderSettings
 *
 * @author Dan Cervelli
 */
public class HelicorderViewerSettings
{
    public String channel;
    public int timeChunk; // seconds
    public int span; // minutes
    public int waveZoomOffset; // seconds
    private double bottomTime; // Double.NaN for now
    public int refreshInterval;
    public int scrollSize;
    public boolean forceCenter;
    public int clipBars;
    private long lastBottomTimeSet;

    public boolean autoScale;
    public boolean showClip;
    public boolean alertClip;
    public int alertClipTimeout;
    public int clipValue;
    public int barRange;
    public double barMult;
    public HelicorderViewPanel view;

    private static SwarmConfig swarmConfig;

    public HelicorderViewerSettings(String ch)
    {
        swarmConfig = SwarmConfig.getInstance();

        channel = ch;
        timeChunk = swarmConfig.timeChunk * 60;
        span = swarmConfig.span * 60;
        waveZoomOffset = 30;
        bottomTime = Double.NaN;
        refreshInterval = 15;
        scrollSize = 24;
    }
}
```

```

forceCenter = false;
clipBars = 21;

clipValue = 2999;
showClip = swarmConfig.showClip;
alertClip = swarmConfig.alertClip;
alertClipTimeout = swarmConfig.alertClipTimeout * 60;
barRange = 1500;
barMult = 3;
autoScale = true;
}

public long getLastBottomTimeSet()
{
    return System.currentTimeMillis() - lastBottomTimeSet;
}

public void setBottomTime(double bt)
{
    lastBottomTimeSet = System.currentTimeMillis();
    bottomTime = bt;
}

public double getBottomTime()
{
    return bottomTime;
}

public void set(ConfigFile cf)
{
    timeChunk = Integer.parseInt(cf.getString("timeChunk"));
    span = Integer.parseInt(cf.getString("span"));
    waveZoomOffset = Integer.parseInt(cf.getString("waveZoomOffset"));
    refreshInterval = Integer.parseInt(cf.getString("refreshInterval"));
    scrollSize = Integer.parseInt(cf.getString("scrollSize"));
    clipValue = Integer.parseInt(cf.getString("clipValue"));
    clipBars = Integer.parseInt(cf.getString("clipBars"));
    barRange = Integer.parseInt(cf.getString("barRange"));
    alertClipTimeout = Integer.parseInt(cf.getString("alertClipTimeout"));
    setBottomTime(Double.parseDouble(cf.getString("bottomTime")));
    barMult = Double.parseDouble(cf.getString("barMult"));
    forceCenter = Boolean.parseBoolean(cf.getString("forceCenter"));
    autoScale = Boolean.parseBoolean(cf.getString("autoScale"));
    showClip = Boolean.parseBoolean(cf.getString("showClip"));
    alertClip = Boolean.parseBoolean(cf.getString("alertClip"));
}

public void save(ConfigFile cf, String prefix)
{
    cf.put(prefix + ".channel", channel);
}

```

```

        cf.put(prefix + ".timeChunk", Integer.toString(timeChunk));
        cf.put(prefix + ".span", Integer.toString(span));
        cf.put(prefix + ".waveZoomOffset", Integer.toString(waveZoomOffset));
        cf.put(prefix + ".refreshInterval", Integer.toString(refreshInterval));
        cf.put(prefix + ".scrollSize", Integer.toString(scrollSize));
        cf.put(prefix + ".clipValue", Integer.toString(clipValue));
        cf.put(prefix + ".clipBars", Integer.toString(clipBars));
        cf.put(prefix + ".barRange", Integer.toString(barRange));
        cf.put(prefix + ".alertClipTimeout", Integer.toString(alertClipTimeout));
        cf.put(prefix + ".bottomTime", Double.toString(bottomTime));
        cf.put(prefix + ".barMult", Double.toString(barMult));
        cf.put(prefix + ".forceCenter", Boolean.toString(forceCenter));
        cf.put(prefix + ".autoScale", Boolean.toString(autoScale));
        cf.put(prefix + ".showClip", Boolean.toString(showClip));
        cf.put(prefix + ".alertClip", Boolean.toString(alertClip));
    }

    public void parseSettingsString(String o)
    {
        // String[] opts = Util.splitString(o, ",");
        String[] opts = o.split(",");
        for (int i = 0; i < opts.length; i++)
        {
            try
            {
                String key = opts[i].substring(0, opts[i].indexOf('='));
                String value = opts[i].substring(opts[i].indexOf('=') + 1);

                if (key.equals("x")) // minutes
                    timeChunk = Integer.parseInt(value) * 60;
                else if (key.equals("y")) // hours
                    span = Integer.parseInt(value) * 60;
            }
            catch (Exception e)
            {
                System.err.println("Could not parse setting: " + opts[i]);
            }
        }
    }

    public void notifyView()
    {
        if (view != null)
            view.settingsChanged();
    }
}

```

2.3.2.3 Fungsi tambahan

```

/*
 * Earthquake Catcher Network v1.0
 * Capstone Design by
 * Christoporus Deo Putratama
 * Kevin Shidqi
 * Bramantio Yuwono
 *
 * Read seismic waves using IMU sensor
 * Read location and exact time using GPS
 * Sending those data using Wi-Fi using MQTT
 *
 */

//=====================================================================
//=====Library & Constant=====
//=====================================================================

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <TinyGPS++.h>

#define SDA_PIN D2
#define SCL_PIN D1
#define PWR_MGMT 0x6B
#define IMU_RES 1
#define RXPin D3
#define TXPin D4
#define GPSBaud 9600
#define SendPeriod 500 //in ms


//=====================================================================
//=====Connection & Database Variables=====
//=====================================================================

const char* ssid = "LSKK Basement"; // network SSID (name)
const char* pass = "noiznocon"; // network password
const char* mqtt_server = "black-boar.rmq.cloudamqp.com"; //MQTT server
const char* server_topic = "amq.topic.ecn"; //MQTT server topic
String mqtt_clientID = "ESP8266Client-1";
String mqtt_user = "lsowqccg:lsowqccg";
String mqtt_password = "kbLv9YbzjQwxz20NH7Rfy98TTV2eK17j";
int status = WL_IDLE_STATUS;
WiFiClient espClient;
PubSubClient client(espClient);
long lstMsg = 0;
int value = 0;

//=====================================================================
//=====IMU & GPS Initiation=====
//=====================================================================

```

```

// I2C address of the MPU-6050
const int MPU=0x68;

struct MPU9255{
    uint8_t x;
    uint8_t y;
    uint8_t z;
};

MPU9255 data;

//TinyGPS++ Object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);

//=====
//=====JSON OBJECT=====
//=====

struct acc {
    String x;
    String y;
    String z;
};

struct geometry{
    String type;
    String coordinates[2];
};

struct prop{
    String Name;
};

struct geojson{
    String type;
    geometry geo;
    prop property;
};

struct data {
    String pointTime;
    String timeZone;
    String interval;
    geojson geometry;
    acc accelerations[20];
};

struct data msg;

struct data InitJsonObject(struct data msg);
String JsonToString(struct data msg);
int i = 0;

//=====
//=====MAIN ALGORITHM=====
//=====

void setup() {

```

```

// put your setup code here, to run once:
pinMode(BUILTIN_LED, OUTPUT);           // Initialize the BUILTIN_LED pin as
an output
MPU9255_Init();
ss.begin(GPSBaud);
Serial.begin(9600);

msg = InitJsonObject(msg);

delay(100);
WiFiConnect();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);

}

void loop() {
// put your main code here, to run repeatedly:

data = Acc_Read();
msg.accelerations[i].x = data.x;
msg.accelerations[i].y = data.y;
msg.accelerations[i].z = data.z;

i++;
if (!client.connected()) {
    reconnect_server();
}
client.loop();
long now = millis();
//if (now - lstMsg > SendPeriod)
if(i==20)
{
    i = 0;
    while (ss.available() > 0)
    if (gps.encode(ss.read()))
        displayInfo();
    else
        Serial.println("INVALID");

//lstMsg = now;
String message = JsonToString(msg);
//msg.printTo(message);
char message_t[800];
message.toCharArray(message_t,800);
//publish sensor data to MQTT broker
bool test = client.publish(server_topic, message_t);
if(test)
    Serial.println("publish success");
}
delay(25);
}

//=====================================================================
//=====Wi-Fi Connection & MQTT Function Procedure=====
//=====================================================================
void WiFiConnect()
{
    // We start by connecting to a WiFi network

```

```

Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
randomSeed(micros());
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void reconnect_server() {
// Loop until we're reconnected
while (!client.connected())
{
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    //if you MQTT broker has clientID,username and password
    //please change following line to if
    (client.connect(clientId,userName,password))
    if (client.connect(mqtt_clientID.c_str(),
mqtt_user.c_str(),mqtt_password.c_str()))
    {
        Serial.println("connected");
    } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        // Wait 6 seconds before retrying
        delay(6000);
    }
}
} //end reconnect()

void callback(char* topic, byte* payload, unsigned int length)
{
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();

    // Switch on the LED if an 1 was received as first character
    if ((char)payload[0] == '1') {
        digitalWrite(BUILTIN_LED, LOW);      // Turn the LED on (Note that LOW
is the voltage level
        // but actually the LED is on; this is because
        // it is active low on the ESP-01)
    } else {
        digitalWrite(BUILTIN_LED, HIGH);     // Turn the LED off by making the
voltage HIGH
    }
}

```

```

}

//=====
//=====IMU & GPS Function Procedure=====
//=====

void MPU9255_Init()
{
    Wire.begin(SDA_PIN, SCL_PIN); // sda, scl
    Wire.beginTransmission(MPU);
    Wire.write(PWR_MGMT); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);

}

MPU9255 Acc_Read()
{
    MPU9255 data;

    Wire.beginTransmission(MPU);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,8,true); // request a total of 6 registers
    data.x=(Wire.read()<<8|Wire.read())*IMU_RES; // 0x3B (ACCEL_XOUT_H) &
    0x3C (ACCEL_XOUT_L)
    data.y=(Wire.read()<<8|Wire.read())*IMU_RES; // 0x3D (ACCEL_YOUT_H) &
    0x3E (ACCEL_YOUT_L)
    data.z=(Wire.read()<<8|Wire.read())*IMU_RES; // 0x3F (ACCEL_ZOUT_H) &
    0x40 (ACCEL_ZOUT_L)

    return data;
}

void displayInfo()
{
    if(gps.location.isValid())
    {
        msg.geometry.geo.coordinates[0] = String(gps.location.lat(),3);
        msg.geometry.geo.coordinates[1] = String(gps.location.lng(),3);

        Serial.println(gps.location.lat(),3);
        Serial.println(gps.location.lng(),3);
    }
    else
    {
        Serial.println("INVALID");
    }

    if(gps.date.isValid() && gps.time.isValid())
    {
        String YEAR = String(gps.date.year());
        String MONTH = String(gps.date.month());
        String DATE = String(gps.date.day());
        String HOUR = String(gps.time.hour());
        String MINUTE = String(gps.time.minute());
        String SECOND = String(gps.time.second());
        msg.pointTime = YEAR + "-" + MONTH + "-" + DATE + "T" + HOUR + ":" +
        MINUTE + ":" + SECOND + "Z";
    }
}

```

```

    }
    else
    {
        Serial.println("INVALID");
    }
}

//=====
//=====ENCODE JSON FUNCTION=====
//=====

struct data InitJsonObject(struct data msg)
{
    msg.pointTime = "test";
    msg.timeZone = "Asia/Jakarta";
    msg.interval = "500";
    msg.geometry.type = "Feature";
    msg.geometry.geo.type = "Point";
    msg.geometry.geo.coordinates[0] = "123.6";
    msg.geometry.geo.coordinates[1] = "123.6";
    msg.geometry.property.Name = "ITB";

    return msg;
}

String JsonToString(struct data msg)
{
    String a = "";

    a = a + "{" + "\"pointTime\": " + "\"" + msg.pointTime + "\"" + "," ;
    a = a + "\"timeZone\": " + "\"" + msg.timeZone + "\"" + "," ;
    a = a + "\"interval\": " + msg.interval + "," ;
    a = a + "\"geojson\":" + "{";

    a = a + "\"type\":" + "\"" + msg.geometry.type + "\"" + "," ;
    a = a + "\"geometry\":" + "{";

    a = a + "\"type\":" + "\"" + msg.geometry.geo.type + "\"" + "," ;
    a = a + "\"coordinates\":[ " + msg.geometry.geo.coordinates[0] + ",";
    a = a + msg.geometry.geo.coordinates[1] + "]";
    a = a + "},";

    a = a + "\"properties\":" + "{";
    a = a + "\"name\":" + "\"" + msg.geometry.property.Name + "\"";
    a = a + "}";

    a = a + "},";
    a = a + "\"accelerations\":[";
    for (int i=0; i<20; i++)
    {
        if(i != 19)
        {
            a = a + "{";
            a = a + "\"x\":" + msg.accelerations[i].x + ",";
            a = a + "\"y\":" + msg.accelerations[i].y + ",";
            a = a + "\"z\":" + msg.accelerations[i].z ;
        }
    }
}

```

```
a = a + "}, ";
}
else
{
    a = a + "}";
a = a + "\"x\": " + msg.accelerations[i].x + ", ";
a = a + "\"y\": " + msg.accelerations[i].y + ", ";
a = a + "\"z\": " + msg.accelerations[i].z ;
a = a + "}";
}

a = a + "]";
a = a + "}";

return a;
}
```



INSTITUT TEKNOLOGI BANDUNG

PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 **(022)2508135-36**, **(022)2500940**
BANDUNG 40132

Dokumentasi Produk Tugas Akhir

Lembar Sampul Dokumen

Judul Dokumen

TUGAS AKHIR TEKNIK ELEKTRO:
Sistem Detektor Gempa dan Tsunami Decision
Support System

Jenis Dokumen

PENGUJIAN

Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB

Nomor Dokumen

B500-02-TA1617.01. 069

Nomor Revisi

01

Nama File

B500

Tanggal Penerbitan

12 May 2017

Unit Penerbit

Prodi Teknik Elektro - ITB

Jumlah Halaman

21

(termasuk lembar sampul ini)

Data Pemeriksaan dan Persetujuan

Ditulis oleh	Nama	Christoporus Putratama	Deo	Jabatan	Anggota
	Tanggal	12 May 2017		Tanda Tangan	
	Nama	Bramantio Yuwono		Jabatan	Anggota
	Tanggal	12 May 2017		Tanda Tangan	
	Nama	Kevin Shidqi Prakoso		Jabatan	Anggota
	Tanggal	12 May 2017		Tanda Tangan	
Disetujui oleh	Nama	Dr. techn. Ary Setijadi Prihatmanto		Jabatan	Dosen Pembimbing
	Tanggal	12 May 2017		Tanda Tangan	

DAFTAR ISI

DAFTAR ISI.....	2
CATATAN SEJARAH PERBAIKAN DOKUMEN	3
PROPOSAL PERANCANGAN SISTEM DETEKTOR GEMPA DAN TSUNAMI DECISION SUPPORT SYSTEM.....	4
1 PENGANTAR	4
1.1 RINGKASAN ISI DOKUMEN	4
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN.....	4
1.3 REFERENSI.....	4
1.4 DAFTAR SINGKATAN	4
2 PENGUJIAN DAN VERIFIKASI PROYEK.....	5
2.1 PENGUJIAN SENSOR SEISMIK ECN (EARTHQUAKE CATCHER NETWORK)	5
2.1.2 <i>Pengujian Sensor IMU 9255</i>	6
2.1.3 <i>Pengujian Algoritma Sensor ECN</i>	38
2.1.4 <i>Pengujian Daya Tahan Sistem Daya</i>	39
2.2 PENGUJIAN MESSAGING SERVER DAN DATABASE.....	41
2.2.1 <i>Pengujian Messaging Server RabbitMQ</i>	41
2.2.2 <i>Pengujian Database MySQL</i>	42
2.2.3 <i>Pengujian Latency Jaringan Sistem</i>	43
2.3 PENGUJIAN ALGORITMA ANALISIS GEMPA	45
2.3.1 <i>Pengujian Algoritma Umum Program</i>	45
2.3.2 <i>Pengujian Algoritma Counting</i>	47
2.3.3 <i>Pengujian Algoritma Magnitude</i>	50
2.3.4 <i>Pengujian Algoritma Spektrum Frekuensi</i>	52
LAMPIRAN	53

Catatan Sejarah Perbaikan Dokumen

VERSI	TANGGAL	OLEH	PERBAIKAN
1	27 April 2017	C. Deo P. Bramantio Y. Kevin S.P.	Versi pertama dokumen B500

Proposal Perancangan Sistem Detektor Gempa dan Tsunami Decision Support System

1 Pengantar

1.1 Ringkasan Isi Dokumen

Dokumen ini berisi tentang perancangan dari Tugas Akhir yang berjudul Sistem Detektor Gempa dan Tsunami Decision Support System Bab 1 berisi pengantar isi dokumen yang terdiri dari ringkasan isi dokumen, tujuan penulisan, referensi, serta daftar singkatan. Pada Bab 2, berisi pengujian modul-modul yang telah dibuat pada B400. Bab ini terdiri dari pengujian sensor gempa, pengujian messaging server dan database, dan pengujian sistem pengolahan data gempa.

1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen

Tujuan penulisan dokumen ini adalah sebagai bahan dokumentasi tahapan pengujian proyek perancangan Sistem Detektor Gempa dan Tsunami Decision Support System, sebagai acuan untuk pihak lain yang ingin mempelajari proyek ini.

1.3 Referensi

- [1] Rudloff, Alexander, *German-Indonesian Tsunami Early Warning System (GITEWS) Decision Support System (DSS)*, Deutschen Zentrums für Luft- und Raumfahrt (DLR): Köln (2010)
- [2] <http://www.codeproject.com/Articles/1117162/Detecting-Seismic-Waves-with-Android-and-IOT>, diakses pada 18 September 2016.

1.4 Daftar Singkatan

SINGKATAN	ARTI
LIPI	LEMBAGA ILMU PENGETAHUAN INDONESIA
GPS	<i>Global Positioning System</i>
IMU	<i>Inertial Measurment Unit</i>
PAU	<i>Pusat Antar Universitas</i>
GUI	<i>Graphics User Interface</i>

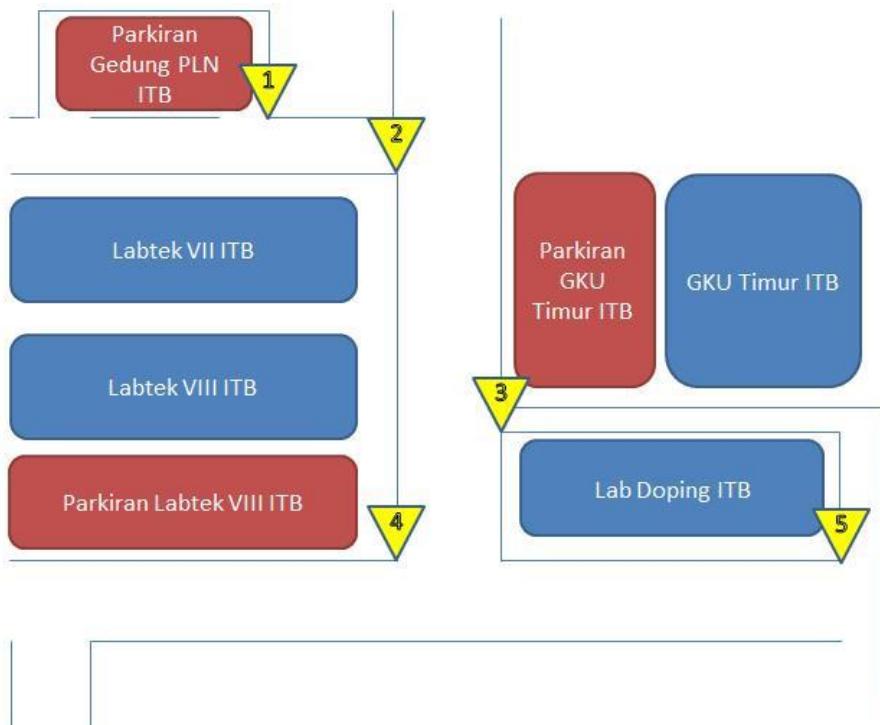
2 Pengujian dan Verifikasi Proyek

2.1 Pengujian Sensor Seismik ECN (Earthquake Catcher Network)

Pengujian Sensor Seismik ECN berfungsi untuk mengetahui performa sensor yang sudah didesain. Pengujian diperlukan untuk melihat performa sensor ECN dari beberapa aspek, yaitu Akurasi GPS, Akurasi IMU, Algoritma Sensor ECN, dan Daya Tahan Sistem Daya.

2.1.1. Pengujian GPS

Pengujian akurasi GPS dilakukan dengan cara mengukur pembacaan GPS pada suatu titik tertentu dengan menggunakan modul GPS Ublox Neo-6m dan dibandingkan dengan pembacaan data longitude dan latitude pada Google Maps di titik yang sama. Pengujian GPS ini dilakukan di beberapa titik di area labtek VIII ITB, Labtek VII ITB, GKU Timur ITB, dan Laboratorium Doping ITB. Berikut adalah titik pengujian GPS yang digunakan.



Gambar 2.1 Denah Titik Tempat Pengujian GPS

Titik pengujian sengaja dilakukan di sudut jalan untuk memudahkan proses pengujian dan mengurangi error yang dilakukan ketika menetapkan titik tersebut pada Google Maps. Berikut adalah perbandingan antara data GPS Google Maps dan modul GPS Ublox Neo 6M.

Titik	Google Maps		Pengujian GPS		Error	
	Lat	Lon	Lat	Lon	Lat	Lon
1	-6,889866	107,610984	-6,889791	107,610939	7,5E-05	4,5E-05
2	-6,889926	107,611537	-6,88995	107,611649	2,4E-05	0,000112
3	-6,890599	107,611615	-6,890493	107,611603	0,000106	1,2E-05
4	-6,890971	107,611534	-6,890993	107,61158	2,2E-05	4,6E-05
5	-6,890958	107,612103	-6,890976	107,612076	1,8E-05	2,7E-05
Error					4,9E-05	4,8E-05

Tabel 2.1 Pengujian GPS

Dapat dilihat pada tabel diatas bahwa error dari pembacaan latitude pada modul GPS Neo 6M adalah 0,000049 dan error dari pembacaan longitude pada modul GPS Neo 6M adalah 0,000048. Pada jalur khatulistiwa, 1 derajat latitude sebanding dengan 110,57 km dan 1 derajat longitude sebanding dengan 111,32 km. Sehingga dapat disimpulkan bahwa error dari pembacaan modul GPS Neo 6M adalah 5,41793 m pada latitude dan 5,34336 m.

2.1.2 Pengujian Sensor IMU 9255

2.1.2.1 Metode Pengujian

Pengujian terhadap sensor IMU 9255 dilakukan dengan cara membandingkan nilai yang terbaca pada sensor dengan pembacaan sensor yang telah terkalibrasi. Pada awalnya, dilakukan perbandingan dengan menggunakan sensor seismik yang dimiliki oleh Divisi Geoteknologi LIPI. Berikut adalah gambar sensor seismik yang dimiliki oleh Divisi Geoteknologi LIPI.



Gambar Sensor Seismik Divisi Geoteknologi LIPI

Tetapi dikarenakan tidak sesuainya data yang dicatat oleh sensor seismik LIPI dengan pembacaan sensor ECN, serta tidak diizinkannya untuk melakukan goyangan terhadap

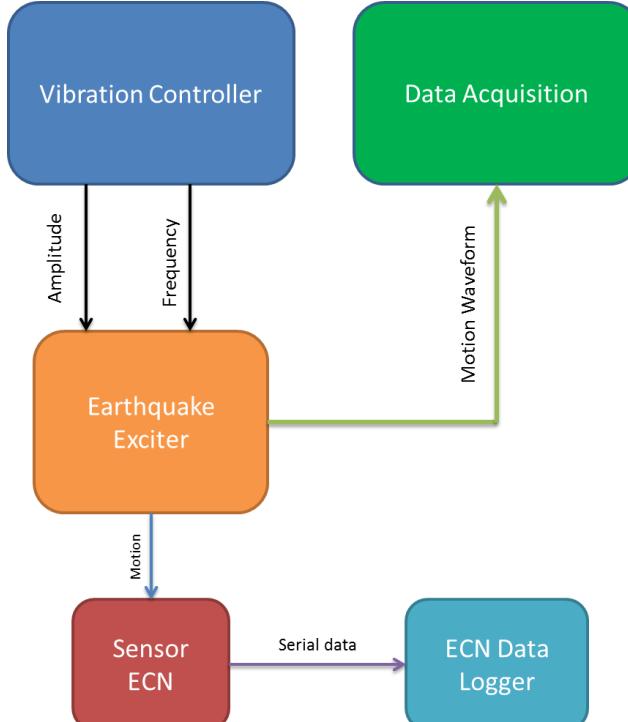
sensor karena ditakutkan rusak maka pengujian tidak dapat dilakukan. Berikut data yang cuplikan data yang dibaca oleh sensor seismik divisi geoteknologi LIPI.

Day	Date	Time	Raw	raw (m/s ²)	Corrected	corr (m/s ²)	Tide	Length	CrossV	AmbTemp	Sensor	Res	Beam	lviCorr	DriftCorr	SPcorr	BattComp	PolarC		
21	338.0035	2016 10 17	0 5 40	128261.1	0.6113	128198.81	-0.6277	14.05	-23.28	23.67	58.95335	911.15	835.4	495.98	0.12	-5660.68	-64.4	-21.55	-1	
3	338.0035	2016 10 17	0 5 40	128261.1	0.6113	128198.81	-0.6019	97.33	14	-23.28	23.67	58.95336	911.14	835.4	495.98	0.12	-5660.69	-64.4	-21.56	-1
4	338.0040	2016 10 17	0 10 40	128260.6	0.6063	128198.81	-0.5779	94.49	13.92	-23.32	23.68	58.95336	911.14	835.4	495.97	0.12	-5660.79	-64.4	-21.56	-1
5	338.0040	2016 10 17	0 10 40	128260.6	0.6063	128198.81	-0.5677	94.49	14.04	-23.37	23.69	58.95336	911.14	835.4	495.97	0.12	-5660.79	-64.4	-21.55	-1
6	338.0049	2016 10 17	0 20 40	128255.7	0.5973	128147.79	-0.5221	88.07	14.09	-23.37	23.69	58.95335	911.24	835.4	495.98	0.12	-5661.31	-64.4	-21.53	-1
7	338.0174	2016 10 17	0 25 40	128259.8	0.5968	128151.32	-0.4886	84.5	14.11	-23.34	23.69	58.95335	911.32	835.4	495.99	0.11	-5661.46	-64.4	-21.5	-1
8	338.0206	2016 10 17	0 30 40	128259.7	0.5968	128151.18	-0.4465	80.7	14.11	-23.1	23.69	58.95335	911.38	835.4	495.98	0.11	-5661.62	-64.4	-21.49	-1
9	338.0206	2016 10 17	0 30 40	128259.7	0.5968	128151.18	-0.4075	76.68	14.07	-23.06	23.69	58.95335	911.41	835.4	495.98	0.11	-5661.76	-64.4	-21.46	-1
10	338.0279	2016 10 17	0 40 40	128255.5	0.5973	128150.53	-0.2971	72.24	14.01	-23.04	23.69	58.95335	911.45	835.4	495.98	0.11	-5661.93	-64.4	-21.47	-1
11	338.0313	2016 10 17	0 45 40	128255.7	0.5973	128150.04	-0.1396	68	13.95	-22.98	23.7	58.95336	911.46	835.4	495.98	0.11	-5662.09	-64.4	-21.46	-1
12	338.0447	2016 10 17	0 50 40	128259.8	0.5979	128172.78	-0.2722	63.36	13.91	-22.52	23.7	58.95336	911.52	835.4	495.98	0.11	-5662.09	-64.4	-21.4	-1
13	338.0482	2016 10 17	0 55 40	128259.8	0.5991	128177.78	-0.2226	58.52	13.95	-22.88	23.7	58.95336	911.59	835.4	495.98	0.11	-5662.4	-64.4	-21.42	-1
14	338.0482	2016 10 17	1 0 40	128260.1	0.6023	128150.1	-0.17	53.51	14.07	-22.87	23.71	58.95336	911.66	835.4	495.98	0.11	-5662.56	-64.4	-21.42	-1
15	338.0491	2016 10 17	1 0 40	128260.1	0.6023	128150.1	-0.1621	53.51	14.07	-22.87	23.71	58.95336	911.66	835.4	495.98	0.11	-5662.56	-64.4	-21.43	-1
16	338.0498	2016 10 17	1 10 40	128260.3	0.6023	128150.3	-0.0615	49.96	13.96	-22.88	23.71	58.95335	911.79	835.4	495.98	0.11	-5662.87	-64.4	-21.36	-1
17	338.0521	2016 10 17	1 15 40	128260.3	0.6027	128150.3	-0.0068	37.45	13.83	-22.95	23.71	58.95335	911.83	835.4	495.98	0.11	-5663.03	-64.4	-21.35	-1
18	338.0556	2016 10 17	1 20 40	128260.2	0.6024	128150.2	0.0596	31.79	13.83	-22.95	23.72	58.95337	911.86	835.4	495.99	0.11	-5663.18	-64.4	-21.34	-1
19	338.0562	2016 10 17	1 25 40	128260.2	0.6024	128150.2	0.1081	26	13.83	-22.96	23.72	58.95337	911.87	835.4	495.99	0.11	-5663.33	-64.4	-21.33	-1
20	338.0562	2016 10 17	1 30 40	128260.2	0.6024	128150.2	0.1656	20.66	13.83	-22.96	23.72	58.95337	911.87	835.4	495.99	0.11	-5663.48	-64.4	-21.33	-1
21	338.0606	2016 10 17	1 35 40	128260.1	0.6008	128224.69	0.2396	14.05	13.57	-22.98	23.73	58.9534	911.87	835.4	495.99	0.11	-5663.85	-64.4	-21.34	-1
22	338.0694	2016 10 17	1 40 40	128260.1	0.6005	128228.78	0.2878	7.91	13.6	-22.95	23.73	58.95339	911.84	835.4	495.98	0.11	-5663.9	-64.4	-21.35	-1
23	338.0729	2016 10 17	1 45 40	128260.1	0.6008	128235.04	0.3504	1.69	13.57	-22.91	23.73	58.95338	911.8	835.4	495.98	0.11	-5663.96	-64.4	-21.36	-1

Gambar Cuplikan Data yang Terbaca pada Sensor Seismik LIPI

Dapat dilihat pada gambar diatas, bahwa data yang terbaca merupakan data yang sudah mengalami pemrosesan, sedangkan sensor ECN hanya dapat memperoleh data mentah percepatan dan pemrosesan dilakukan pada server sehingga tidak dapat dilakukan karena latensi jaringan yang tidak dapat diprediksi sehingga data dari kedua buah sensor sulit untuk dibandingkan.

Sebagai pengganti sensor seismik Divisi Geoteknologi LIPI, digunakan Earthquake Exciter Lab Dinamika PAU ITB sebagai sensor pembanding yang sudah terkalibrasi. Untuk skema pengujian dapat dilihat pada gambar berikut.



Gambar Skema Pengujian Sensor IMU MPU9255 menggunakan Earthquake Exciter

Earthquake exciter adalah sebuah alat yang dapat menghasilkan getaran sesuai dengan input magnitude dan frekuensi yang diberikan. Pada pengujian ini sensor IMU MPU9255 diletakkan dan ditempelkan pada Earthquake Exciter sehingga IMU akan bergetar bersamaan dengan Earthquake Exciter. Input frekuensi dan amplitudo diberikan dengan menggunakan Vibration Controller yang terhubung dengan Earthquake Exciter. Data acquisition digunakan untuk memvalidasi besar getaran yang dihasilkan oleh Earthquake Exciter. Berikut adalah gambar beberapa alat yang digunakan dalam melakukan pengujian sensor IMU MPU9255.



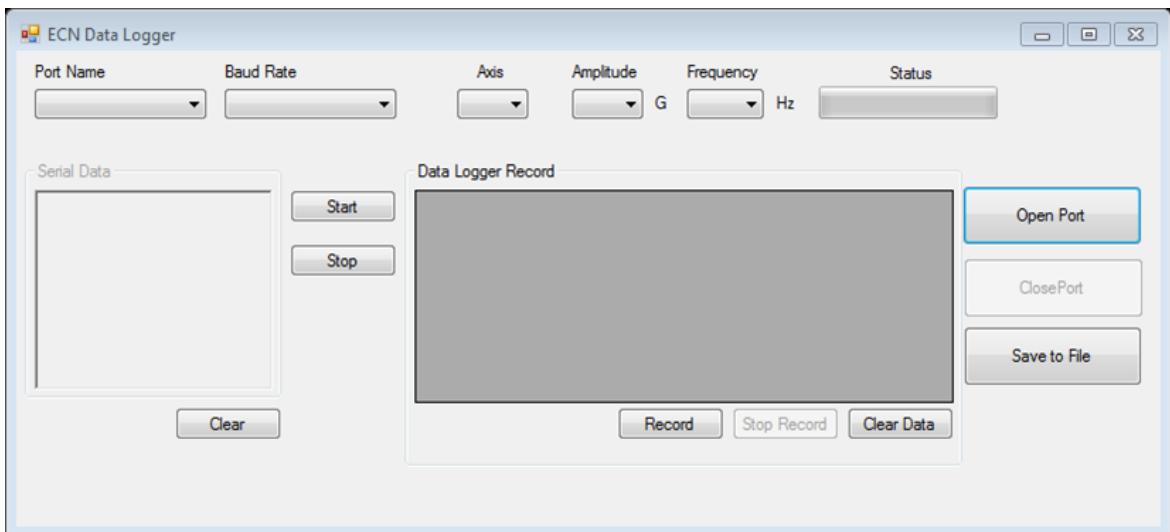
Gambar Earthquake Exciter (kiri), Vibration Controller (tengah), Data Aquisition (kanan)

Pengujian dilakukan pada amplitudo 0.5 G, 1.0 G, dan 1.5 G untuk setiap frekuensi 10 Hz, 15 Hz, 20 Hz dan sumbu x, y, dan z. Sehingga dilakukan pengujian sebanyak 27 kali. Frekuensi dan Amplitudo dapat diubah menggunakan vibration controller, sedangkan pengujian untuk sumbu x,y, dan z dilakukan dengan mengubah posisi peletakan/penempelan sensor IMU MPU9255 pada Earthquake Exciter. Gambar peletakan sensor IMU MPU9255 pada sumbu x,y,z dapat dilihat pada gambar berikut.



Gambar Peletakan Sensor IMU MPU9255 untuk Pengujian Sumbu x (tengah), y (kiri), z (kanan)

Untuk memudahkan proses pembacaan data sensor ECN, sensor MPU9255 terhubung ke sebuah NodeMCU yang selalu membaca percepatan dengan frekuensi 40 Hz dan mengirimkan hasil pembacaan tersebut ke GUI ECN Data Logger dengan menggunakan komunikasi serial.



Gambar Tampilan GUI ECN Data Logger

Langkah-langkah untuk menggunakan GUI ini adalah sebagai berikut.

1. Pilih Port Name dan BaudRate sesuai dengan yang digunakan, lalu klik Open Port
2. Setelah port serial berhasil dibuka, tekan tombol Start untuk memulai mengambil data yang dikirimkan dari NodeMCU, tombol stop untuk menghentikan proses mengambil data yang dikirimkan dari NodeMCU
3. Tombol Record digunakan untuk memulai mencatat pembacaan data dari String yang dikirimkan oleh NodeMCU ke dalam tabel, Clear Data untuk menghapus tabel yang sudah direkam, dan Stop Record untuk menghentikan proses penyimpanan data serial dari NodeMCU ke tabel.
4. Save to File digunakan untuk menyimpan tabel yang sudah dicatat ke dalam file excel

Open Port dapat dilakukan dengan menggunakan fungsi SerialPort.Open() dan Close Port dapat diimplementasikan dengan menggunakan fungsi SerialPort.Close(). Proses pembacaan string dilakukan dengan menggunakan timer setiap 1ms dengan menggunakan SerialPort.ReadLine(). Ketika tombol Record ditekan, proses penyimpanan data pada tabel akan mulai dilakukan dengan assign variabel boolean dengan nilai true. Sedangkan untuk perintah Save to File dapat dilihat dari potongan source code sebagai berikut.

```

SaveFileDialog sfd = new SaveFileDialog();
sfd.Filter = "Excel Documents (*.xlsx)|*.xlsx";
String name = comboBox1.SelectedItem.ToString() + " " +
AmplitudeBox.SelectedItem.ToString() + "G " +
FrequencyBox.SelectedItem.ToString()+"Hz";
sfd.FileName = "Test MPU9255 " + name +".xlsx";
if (sfd.ShowDialog() == DialogResult.OK)
{
    var workbook = new XLWorkbook();
    workbook.Worksheets.Add(table, "Test MPU9255");
    workbook.SaveAs(sfd.FileName);
    table.Clear();
    MessageBox.Show("Save File Success !", "Information", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

```

2.1.2.2 Hasil Pengujian

2.1.2.2.1 Tes Sumbu X pada 1,0 G

	10 Hz			15 Hz			20 Hz	
Acc X	Acc Y	Acc Z	Acc X2	Acc Y3	Acc Z4	Acc X3	Acc Y4	Acc Z5
-0,9748	0,0250	0,0269	-0,9703	0,0270	0,0272	-0,9777	0,0221	0,0254
-0,9901	0,0245	0,0275	-0,9809	0,0259	0,0265	-0,9660	0,0224	0,0252
-0,9795	0,0246	0,0269	-0,9826	0,0259	0,0268	-0,9774	0,0220	0,0258
-0,9643	0,0249	0,0265	-0,9706	0,0261	0,0267	-0,9671	0,0224	0,0257
-0,9746	0,0244	0,0260	-0,9865	0,0256	0,0265	-0,9784	0,0222	0,0263
-0,9897	0,0239	0,0265	-0,9762	0,0256	0,0265	-0,9669	0,0226	0,0266
-0,9796	0,0240	0,0266	-0,9748	0,0307	0,0234	-0,9781	0,0225	0,0263
-0,9639	0,0243	0,0268	-0,9865	0,0303	0,0234	-0,9664	0,0228	0,0263
-0,9741	0,0236	0,0264	-0,9701	0,0306	0,0231	-0,9777	0,0226	0,0264
-0,9888	0,0233	0,0265	-0,9808	0,0301	0,0237	-0,9660	0,0229	0,0265
-0,9789	0,0232	0,0262	-0,9809	0,0300	0,0241	-0,9775	0,0227	0,0266
-0,9636	0,0235	0,0259	-0,9690	0,0297	0,0240	-0,9678	0,0227	0,0265
-0,9742	0,0227	0,0253	-0,9854	0,0290	0,0236	-0,9778	0,0225	0,0265
-0,9892	0,0222	0,0255	-0,9741	0,0295	0,0236	-0,9682	0,0229	0,0262
-0,9789	0,0224	0,0250	-0,9725	0,0316	0,0254	-0,9774	0,0226	0,0264
-0,9632	0,0227	0,0248	-0,9845	0,0313	0,0255	-0,9675	0,0231	0,0264
-0,9743	0,0229	0,0251	-0,9685	0,0313	0,0254	-0,9771	0,0229	0,0262
-0,9893	0,0223	0,0255	-0,9796	0,0304	0,0244	-0,9677	0,0230	0,0259
-0,9791	0,0224	0,0252	-0,9810	0,0304	0,0245	-0,9768	0,0226	0,0260
-0,9637	0,0229	0,0248	-0,9686	0,0305	0,0245	-0,9673	0,0230	0,0257
-0,9745	0,0226	0,0244	-0,9847	0,0299	0,0247	-0,9766	0,0227	0,0258
-0,9895	0,0221	0,0249	-0,9739	0,0300	0,0246	-0,9669	0,0231	0,0259
-0,9791	0,0224	0,0247	-0,9748	0,0320	0,0171	-0,9763	0,0228	0,0259
-0,9636	0,0228	0,0244	-0,9865	0,0312	0,0176	-0,9670	0,0233	0,0255
-0,9746	0,0231	0,0244	-0,9701	0,0312	0,0177	-0,9772	0,0230	0,0260
-0,9896	0,0227	0,0248	-0,9807	0,0309	0,0184	-0,9686	0,0232	0,0264
-0,9790	0,0229	0,0249	-0,9808	0,0304	0,0186	-0,9784	0,0228	0,0260
-0,9630	0,0234	0,0243	-0,9699	0,0305	0,0181	-0,9693	0,0230	0,0258
-0,9736	0,0230	0,0239	-0,9859	0,0303	0,0184	-0,9793	0,0230	0,0256
-0,9886	0,0225	0,0240	-0,9746	0,0305	0,0188	-0,9691	0,0233	0,0255
-0,9785	0,0225	0,0240	-0,9760	0,0290	0,0188	-0,9788	0,0229	0,0257
-0,9631	0,0231	0,0237	-0,9861	0,0286	0,0193	-0,9685	0,0230	0,0257
-0,9737	0,0228	0,0237	-0,9701	0,0287	0,0194	-0,9782	0,0232	0,0257
-0,9885	0,0225	0,0242	-0,9828	0,0291	0,0202	-0,9677	0,0234	0,0261
-0,9772	0,0230	0,0244	-0,9808	0,0290	0,0203	-0,9778	0,0230	0,0261
-0,9623	0,0232	0,0237	-0,9714	0,0287	0,0205	-0,9690	0,0245	0,0258
-0,9735	0,0235	0,0240	-0,9873	0,0287	0,0210	-0,9787	0,0243	0,0254
-0,9878	0,0231	0,0244	-0,9746	0,0288	0,0208	-0,9686	0,0244	0,0252

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 10 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

-0,9764	0,0236	0,0243	-0,9767	0,0303	0,0211	-0,9780	0,0239	0,0248
-0,9624	0,0238	0,0247	-0,9869	0,0297	0,0214	-0,9677	0,0243	0,0250
-0,9747	0,0234	0,0249	-0,9708	0,0296	0,0214	-0,9779	0,0241	0,0247
-0,9886	0,0232	0,0252	-0,9841	0,0297	0,0220	-0,9689	0,0244	0,0245
-0,9765	0,0236	0,0254	-0,9821	0,0296	0,0219	-0,9789	0,0241	0,0244
-0,9626	0,0238	0,0251	-0,9720	0,0296	0,0223	-0,9689	0,0244	0,0247
-0,9744	0,0237	0,0256	-0,9880	0,0290	0,0225	-0,9789	0,0242	0,0250
-0,9887	0,0236	0,0259	-0,9753	0,0290	0,0227	-0,9686	0,0246	0,0247
-0,9771	0,0242	0,0252	-0,9775	0,0280	0,0215	-0,9793	0,0244	0,0250
-0,9631	0,0241	0,0248	-0,9876	0,0276	0,0216	-0,9691	0,0247	0,0250
-0,9751	0,0237	0,0259	-0,9715	0,0277	0,0215	-0,9786	0,0245	0,0250
-0,9891	0,0236	0,0262	-0,9841	0,0278	0,0223	-0,9684	0,0249	0,0251
-0,9768	0,0239	0,0261	-0,9813	0,0278	0,0221	-0,9779	0,0247	0,0251
-0,9629	0,0243	0,0259	-0,9714	0,0280	0,0216	-0,9675	0,0249	0,0250
-0,9753	0,0241	0,0264	-0,9874	0,0277	0,0220	-0,9773	0,0246	0,0253
-0,9888	0,0240	0,0267	-0,9744	0,0278	0,0220	-0,9677	0,0248	0,0249
-0,9771	0,0247	0,0268	-0,9767	0,0282	0,0234	-0,9770	0,0246	0,0249
-0,9633	0,0252	0,0267	-0,9859	0,0279	0,0235	-0,9670	0,0250	0,0248
-0,9754	0,0249	0,0269	-0,9698	0,0280	0,0229	-0,9767	0,0248	0,0246
-0,9893	0,0247	0,0270	-0,9824	0,0274	0,0226	-0,9664	0,0251	0,0243
-0,9773	0,0250	0,0272	-0,9802	0,0278	0,0225	-0,9758	0,0246	0,0242
-0,9630	0,0253	0,0267	-0,9704	0,0278	0,0225	-0,9659	0,0248	0,0241
-0,9752	0,0252	0,0270	-0,9866	0,0275	0,0232	-0,9754	0,0247	0,0243
-0,9895	0,0247	0,0270	-0,9734	0,0277	0,0231	-0,9657	0,0249	0,0242
-0,9770	0,0251	0,0267	-0,9759	0,0263	0,0223	-0,9752	0,0243	0,0243
-0,9632	0,0253	0,0265	-0,9861	0,0261	0,0227	-0,9651	0,0245	0,0241
-0,9760	0,0252	0,0266	-0,9699	0,0263	0,0227	-0,9743	0,0240	0,0246
-0,9901	0,0251	0,0268	-0,9829	0,0261	0,0237	-0,9648	0,0242	0,0246
-0,9781	0,0254	0,0265	-0,9814	0,0262	0,0240	-0,9752	0,0240	0,0246
-0,9635	0,0254	0,0267	-0,9712	0,0264	0,0238	-0,9652	0,0244	0,0246
-0,9758	0,0250	0,0265	-0,9869	0,0263	0,0246	-0,9749	0,0243	0,0246
-0,9898	0,0246	0,0265	-0,9745	0,0266	0,0245	-0,9654	0,0244	0,0243
-0,9783	0,0248	0,0264	-0,9767	0,0276	0,0255	-0,9761	0,0240	0,0245
-0,9637	0,0249	0,0260	-0,9871	0,0272	0,0262	-0,9667	0,0241	0,0243
-0,9767	0,0241	0,0261	-0,9710	0,0273	0,0260	-0,9769	0,0240	0,0248
-0,9905	0,0239	0,0268	-0,9834	0,0267	0,0253	-0,9675	0,0242	0,0249
-0,9779	0,0239	0,0263	-0,9816	0,0268	0,0258	-0,9778	0,0239	0,0248
-0,9639	0,0242	0,0258	-0,9720	0,0267	0,0254	-0,9685	0,0240	0,0247
-0,9763	0,0238	0,0262	-0,9879	0,0262	0,0257	-0,9788	0,0237	0,0247
-0,9902	0,0236	0,0265	-0,9749	0,0264	0,0256	-0,9686	0,0241	0,0243
-0,9782	0,0241	0,0264	-0,9770	0,0251	0,0245	-0,9779	0,0238	0,0246
-0,9642	0,0240	0,0262	-0,9870	0,0251	0,0245	-0,9676	0,0241	0,0247
-0,9761	0,0235	0,0262	-0,9708	0,0254	0,0242	-0,9776	0,0240	0,0250
-0,9900	0,0233	0,0265	-0,9839	0,0249	0,0247	-0,9689	0,0242	0,0250

-0,9779	0,0236	0,0264	-0,9824	0,0252	0,0249	-0,9786	0,0238	0,0249
-0,9639	0,0238	0,0265	-0,9726	0,0253	0,0249	-0,9695	0,0254	0,0251
-0,9769	0,0233	0,0266	-0,9883	0,0253	0,0256	-0,9795	0,0253	0,0252
-0,9900	0,0231	0,0270	-0,9751	0,0255	0,0251	-0,9696	0,0254	0,0250
-0,9772	0,0233	0,0271	-0,9775	0,0248	0,0229	-0,9801	0,0252	0,0250
-0,9644	0,0236	0,0270	-0,9872	0,0245	0,0230	-0,9708	0,0259	0,0252
-0,9779	0,0228	0,0272	-0,9711	0,0247	0,0225	-0,9811	0,0257	0,0255
-0,9913	0,0228	0,0274	-0,9840	0,0248	0,0224	-0,9720	0,0259	0,0253
-0,9786	0,0232	0,0269	-0,9822	0,0246	0,0225	-0,9817	0,0258	0,0255
-0,9655	0,0236	0,0267	-0,9725	0,0248	0,0222	-0,9724	0,0277	0,0254
-0,9787	0,0231	0,0270	-0,9887	0,0247	0,0227	-0,9823	0,0271	0,0253
-0,9916	0,0228	0,0270	-0,9757	0,0249	0,0223	-0,9755	0,0282	0,0224
-0,9787	0,0230	0,0266	-0,9788	0,0245	0,0218	-0,9840	0,0281	0,0226
-0,9656	0,0233	0,0259	-0,9878	0,0244	0,0221	-0,9766	0,0238	0,0205
-0,9789	0,0228	0,0261	-0,9717	0,0250	0,0222	-0,9845	0,0236	0,0205
-0,9908	0,0224	0,0262	-0,9843	0,0256	0,0235	-0,9792	0,0192	0,0145
-0,9769	0,0229	0,0260	-0,9826	0,0256	0,0233	-0,9872	0,0191	0,0153
-0,9643	0,0232	0,0257	-0,9730	0,0254	0,0229	-0,9774	0,0164	0,0174
-0,9786	0,0227	0,0260	-0,9893	0,0253	0,0232	-0,9854	0,0162	0,0173
-0,9911	0,0225	0,0262	-0,9762	0,0254	0,0231	-0,9764	0,0179	0,0179
-0,9771	0,0227	0,0258	-0,9780	0,0264	0,0217	-0,9837	0,0179	0,0187
-0,9644	0,0232	0,0257	-0,9876	0,0263	0,0221	-0,9753	0,0197	0,0178
-0,9786	0,0228	0,0260	-0,9719	0,0266	0,0218	-0,9826	0,0197	0,0182
-0,9911	0,0222	0,0262	-0,9843	0,0263	0,0213	-0,9745	0,0223	0,0184
-0,9770	0,0225	0,0260	-0,9820	0,0266	0,0217	-0,9819	0,0221	0,0189
-0,9642	0,0229	0,0259	-0,9717	0,0265	0,0216	-0,9758	0,0190	0,0142
-0,9786	0,0225	0,0262	-0,9878	0,0260	0,0221	-0,9842	0,0189	0,0149
-0,9912	0,0224	0,0262	-0,9749	0,0261	0,0223	-0,9734	0,0222	0,0185
-0,9769	0,0227	0,0261	-0,9777	0,0248	0,0214	-0,9822	0,0219	0,0192
-0,9645	0,0231	0,0255	-0,9875	0,0245	0,0219	-0,9743	0,0229	0,0176
-0,9782	0,0225	0,0255	-0,9712	0,0248	0,0213	-0,9813	0,0228	0,0180
-0,9905	0,0222	0,0260	-0,9838	0,0248	0,0221	-0,9754	0,0231	0,0188
-0,9769	0,0224	0,0257	-0,9814	0,0249	0,0224	-0,9818	0,0228	0,0189
-0,9647	0,0224	0,0256	-0,9714	0,0252	0,0220	-0,9762	0,0225	0,0189
-0,9784	0,0218	0,0259	-0,9872	0,0247	0,0222	-0,9801	0,0220	0,0197
-0,9909	0,0213	0,0259	-0,9744	0,0249	0,0222	-0,9777	0,0225	0,0203
-0,9763	0,0217	0,0262	-0,9767	0,0266	0,0237	-0,9794	0,0226	0,0207
-0,9635	0,0221	0,0256	-0,9871	0,0263	0,0239	-0,9768	0,0224	0,0213
-0,9775	0,0219	0,0257	-0,9705	0,0266	0,0238	-0,9802	0,0223	0,0215
-0,9904	0,0215	0,0259	-0,9836	0,0264	0,0237	-0,9775	0,0228	0,0221
-0,9760	0,0218	0,0260	-0,9812	0,0264	0,0235	-0,9786	0,0225	0,0225
-0,9639	0,0222	0,0256	-0,9712	0,0264	0,0232	-0,9755	0,0225	0,0231
-0,9776	0,0215	0,0254	-0,9872	0,0258	0,0236	-0,9774	0,0226	0,0235
-0,9900	0,0211	0,0255	-0,9740	0,0262	0,0233	-0,9750	0,0232	0,0240

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 12 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

-0,9768	0,0214	0,0253	-0,9767	0,0246	0,0222	-0,9775	0,0232	0,0239
-0,9643	0,0216	0,0250	-0,9864	0,0246	0,0226	-0,9755	0,0230	0,0243
-0,9783	0,0210	0,0247	-0,9703	0,0246	0,0222	-0,9789	0,0229	0,0245
-0,9907	0,0207	0,0251	-0,9825	0,0252	0,0232	-0,9765	0,0232	0,0246
-0,9765	0,0211	0,0250	-0,9804	0,0252	0,0233	-0,9783	0,0232	0,0251
-0,9637	0,0213	0,0248	-0,9709	0,0253	0,0232	-0,9757	0,0234	0,0256
-0,9779	0,0209	0,0256	-0,9870	0,0251	0,0236	-0,9793	0,0232	0,0260
-0,9905	0,0206	0,0262	-0,9747	0,0255	0,0238	-0,9768	0,0237	0,0258
-0,9765	0,0211	0,0257	-0,9765	0,0278	0,0254	-0,9789	0,0234	0,0261
-0,9642	0,0215	0,0251	-0,9856	0,0277	0,0250	-0,9759	0,0243	0,0266
-0,9780	0,0214	0,0251	-0,9697	0,0278	0,0250	-0,9790	0,0243	0,0265
-0,9905	0,0211	0,0255	-0,9832	0,0280	0,0250	-0,9774	0,0244	0,0272
-0,9768	0,0215	0,0257	-0,9789	0,0278	0,0250	-0,9806	0,0244	0,0271
-0,9643	0,0220	0,0258	-0,9701	0,0280	0,0248	-0,9783	0,0241	0,0273
-0,9781	0,0217	0,0256	-0,9854	0,0274	0,0250	-0,9813	0,0238	0,0275
-0,9908	0,0214	0,0257	-0,9716	0,0275	0,0248	-0,9792	0,0243	0,0277
-0,9767	0,0219	0,0253	-0,9758	0,0284	0,0251	-0,9822	0,0241	0,0278
-0,9636	0,0223	0,0247	-0,9844	0,0280	0,0250	-0,9802	0,0240	0,0282
-0,9775	0,0215	0,0250	-0,9689	0,0281	0,0251	-0,9828	0,0237	0,0283
-0,9894	0,0214	0,0255	-0,9829	0,0273	0,0246	-0,9808	0,0244	0,0281
-0,9754	0,0215	0,0250	-0,9796	0,0273	0,0244	-0,9838	0,0243	0,0280
-0,9628	0,0221	0,0246	-0,9705	0,0275	0,0243	-0,9816	0,0240	0,0279
-0,9771	0,0218	0,0248	-0,9859	0,0272	0,0244	-0,9841	0,0237	0,0282
-0,9896	0,0215	0,0251	-0,9723	0,0274	0,0243	-0,9819	0,0242	0,0286
-0,9755	0,0217	0,0248	-0,9765	0,0258	0,0247	-0,9848	0,0243	0,0288
-0,9630	0,0220	0,0243	-0,9849	0,0259	0,0248	-0,9820	0,0247	0,0291
-0,9773	0,0215	0,0244	-0,9690	0,0261	0,0249	-0,9853	0,0247	0,0288
-0,9895	0,0215	0,0250	-0,9824	0,0259	0,0251	-0,9833	0,0256	0,0292
-0,9760	0,0217	0,0248	-0,9787	0,0260	0,0248	-0,9867	0,0255	0,0291
-0,9635	0,0222	0,0245	-0,9697	0,0260	0,0249	-0,9838	0,0261	0,0292
-0,9776	0,0218	0,0250	-0,9848	0,0262	0,0247	-0,9852	0,0259	0,0290
-0,9897	0,0214	0,0250	-0,9707	0,0265	0,0250	-0,9815	0,0269	0,0296
-0,9761	0,0217	0,0248	-0,9759	0,0267	0,0260	-0,9839	0,0266	0,0292
-0,9636	0,0222	0,0245	-0,9824	0,0262	0,0258	-0,9809	0,0268	0,0297
-0,9770	0,0213	0,0245	-0,9671	0,0264	0,0261	-0,9827	0,0266	0,0295
-0,9893	0,0210	0,0249	-0,9817	0,0262	0,0259	-0,9791	0,0278	0,0305
-0,9749	0,0215	0,0246	-0,9766	0,0263	0,0259	-0,9813	0,0274	0,0301
-0,9632	0,0220	0,0239	-0,9701	0,0290	0,0261	-0,9780	0,0282	0,0305
-0,9778	0,0221	0,0243	-0,9847	0,0285	0,0260	-0,9805	0,0279	0,0307
-0,9892	0,0222	0,0245	-0,9693	0,0286	0,0258	-0,9776	0,0290	0,0313
-0,9744	0,0226	0,0254	-0,9755	0,0291	0,0250	-0,9800	0,0288	0,0316
-0,9622	0,0226	0,0249	-0,9806	0,0286	0,0254	-0,9772	0,0291	0,0317
-0,9771	0,0225	0,0251	-0,9661	0,0289	0,0252	-0,9790	0,0291	0,0316
-0,9882	0,0223	0,0254	-0,9817	0,0282	0,0251	-0,9755	0,0301	0,0321

-0,9732	0,0224	0,0252	-0,9753	0,0281	0,0255	-0,9781	0,0299	0,0320
-0,9614	0,0227	0,0249	-0,9703	0,0280	0,0256	-0,9763	0,0295	0,0319
-0,9765	0,0228	0,0258	-0,9848	0,0275	0,0259	-0,9784	0,0291	0,0320
-0,9884	0,0225	0,0261	-0,9693	0,0276	0,0255	-0,9753	0,0299	0,0327
-0,9739	0,0230	0,0252	-0,9761	0,0272	0,0253	-0,9779	0,0299	0,0325
-0,9618	0,0232	0,0251	-0,9813	0,0267	0,0257	-0,9761	0,0293	0,0326
-0,9768	0,0231	0,0260	-0,9664	0,0270	0,0257	-0,9784	0,0293	0,0319
-0,9886	0,0228	0,0266	-0,9819	0,0265	0,0263	-0,9765	0,0300	0,0323
-0,9742	0,0228	0,0264	-0,9745	0,0265	0,0262	-0,9790	0,0296	0,0321
-0,9625	0,0232	0,0264	-0,9702	0,0277	0,0282	-0,9764	0,0298	0,0321
-0,9777	0,0231	0,0265	-0,9842	0,0274	0,0282	-0,9782	0,0297	0,0313
-0,9895	0,0230	0,0266	-0,9685	0,0277	0,0278	-0,9749	0,0306	0,0308
-0,9748	0,0235	0,0261	-0,9772	0,0286	0,0289	-0,9774	0,0305	0,0312
-0,9632	0,0239	0,0262	-0,9788	0,0285	0,0283	-0,9755	0,0301	0,0314
-0,9777	0,0239	0,0269	-0,9670	0,0285	0,0282	-0,9776	0,0298	0,0307
-0,9887	0,0236	0,0270	-0,9833	0,0279	0,0282	-0,9751	0,0300	0,0313
-0,9744	0,0237	0,0273	-0,9734	0,0281	0,0276	-0,9778	0,0295	0,0310
-0,9632	0,0241	0,0271	-0,9737	0,0155	0,0207	-0,9743	0,0306	0,0316
-0,9782	0,0240	0,0278	-0,9853	0,0156	0,0208	-0,9765	0,0302	0,0316
-0,9893	0,0236	0,0282	-0,9691	0,0161	0,0208	-0,9748	0,0301	0,0320
-0,9747	0,0240	0,0279	-0,9801	0,0161	0,0212	-0,9773	0,0298	0,0318
-0,9631	0,0240	0,0274	-0,9816	0,0163	0,0213	-0,9743	0,0301	0,0321
-0,9776	0,0240	0,0280	-0,9689	0,0169	0,0215	-0,9765	0,0299	0,0321
-0,9888	0,0239	0,0283	-0,9853	0,0168	0,0216	-0,9746	0,0303	0,0323
-0,9739	0,0244	0,0281	-0,9741	0,0171	0,0216	-0,9776	0,0299	0,0316
-0,9623	0,0245	0,0277	-0,9725	0,0193	0,0230	-0,9760	0,0297	0,0320
-0,9771	0,0240	0,0276	-0,9849	0,0191	0,0239	-0,9778	0,0292	0,0320
-0,9882	0,0236	0,0281	-0,9688	0,0197	0,0238	-0,9753	0,0293	0,0323
-0,9733	0,0242	0,0277	-0,9798	0,0195	0,0232	-0,9775	0,0289	0,0319
-0,9620	0,0243	0,0273	-0,9813	0,0197	0,0235	-0,9745	0,0291	0,0312

2.1.2.2 Tes Sumbu X pada 1,5 G

10 Hz			15 Hz			20 Hz		
Acc X	Acc Y	Acc Z	Acc X2	Acc Y3	Acc Z4	Acc X3	Acc Y4	Acc Z5
0,0255	0,7833	0,6675	-0,9534	0,0342	0,0316	-0,9898	0,0328	0,0330
0,0255	0,7833	0,6673	-0,9803	0,0325	0,0313	-0,9749	0,0329	0,0324
0,0254	0,7832	0,6674	-1,0072	0,0316	0,0319	-0,9622	0,0329	0,0324
0,0257	0,7833	0,6675	-0,9805	0,0319	0,0312	-0,9936	0,0312	0,0327
0,0256	0,7833	0,6678	-0,9509	0,0323	0,0308	-0,9607	0,0317	0,0319
0,0256	0,7833	0,6679	-0,9789	0,0302	0,0306	-0,9760	0,0347	0,0331
0,0255	0,7831	0,6679	-1,0063	0,0295	0,0308	-0,9873	0,0342	0,0332
0,0256	0,7833	0,6681	-0,9807	0,0296	0,0304	-0,9554	0,0346	0,0326
0,0256	0,7832	0,6680	-0,9518	0,0301	0,0301	-0,9889	0,0348	0,0333

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 14 dari 58

0,0254	0,7829	0,6681	-0,9780	0,0262	0,0298	-0,9744	0,0346	0,0329
0,0251	0,7831	0,6681	-1,0054	0,0257	0,0301	-0,9633	0,0347	0,0327
0,0249	0,7830	0,6677	-0,9805	0,0265	0,0299	-0,9936	0,0335	0,0329
0,0250	0,7828	0,6680	-0,9521	0,0270	0,0296	-0,9588	0,0340	0,0324
0,0251	0,7828	0,6680	-0,9802	0,0251	0,0293	-0,9757	0,0337	0,0345
0,0252	0,7828	0,6680	-1,0083	0,0247	0,0299	-0,9855	0,0329	0,0340
0,0251	0,7827	0,6680	-0,9826	0,0251	0,0293	-0,9555	0,0333	0,0337
0,0252	0,7829	0,6677	-0,9527	0,0258	0,0284	-0,9899	0,0320	0,0341
0,0254	0,7831	0,6679	-0,9800	0,0246	0,0285	-0,9717	0,0322	0,0338
0,0254	0,7830	0,6679	-1,0065	0,0240	0,0294	-0,9623	0,0321	0,0336
0,0255	0,7827	0,6676	-0,9805	0,0243	0,0293	-0,9917	0,0310	0,0335
0,0253	0,7826	0,6677	-0,9509	0,0248	0,0281	-0,9562	0,0313	0,0334
0,0255	0,7826	0,6676	-0,9786	0,0232	0,0282	-0,9787	0,0293	0,0317
0,0254	0,7827	0,6676	-1,0066	0,0228	0,0289	-0,9862	0,0287	0,0315
0,0254	0,7829	0,6674	-0,9800	0,0237	0,0285	-0,9576	0,0294	0,0311
0,0255	0,7829	0,6673	-0,9507	0,0247	0,0274	-0,9923	0,0276	0,0307
0,0257	0,7829	0,6669	-0,9775	0,0245	0,0273	-0,9705	0,0283	0,0304
0,0255	0,7828	0,6667	-1,0043	0,0240	0,0279	-0,9645	0,0281	0,0303
0,0254	0,7827	0,6665	-0,9788	0,0249	0,0272	-0,9925	0,0271	0,0303
0,0254	0,7826	0,6666	-0,9504	0,0253	0,0270	-0,9574	0,0277	0,0297
0,0256	0,7826	0,6669	-0,9777	0,0235	0,0276	-0,9792	0,0234	0,0273
0,0252	0,7826	0,6667	-1,0059	0,0233	0,0280	-0,9850	0,0234	0,0275
0,0252	0,7825	0,6670	-0,9793	0,0234	0,0277	-0,9556	0,0241	0,0274
0,0253	0,7824	0,6669	-0,9503	0,0238	0,0274	-0,9904	0,0232	0,0272
0,0253	0,7828	0,6671	-0,9786	0,0210	0,0264	-0,9715	0,0238	0,0273
0,0253	0,7828	0,6669	-1,0062	0,0205	0,0265	-0,9655	0,0238	0,0272
0,0253	0,7828	0,6670	-0,9799	0,0211	0,0260	-0,9933	0,0232	0,0273
0,0251	0,7826	0,6668	-0,9510	0,0219	0,0256	-0,9580	0,0241	0,0269
0,0252	0,7826	0,6665	-0,9795	0,0186	0,0253	-0,9801	0,0205	0,0226
0,0253	0,7824	0,6663	-1,0067	0,0185	0,0254	-0,9853	0,0205	0,0222
0,0254	0,7825	0,6663	-0,9817	0,0193	0,0257	-0,9573	0,0213	0,0220
0,0252	0,7827	0,6663	-0,9523	0,0200	0,0253	-0,9921	0,0212	0,0226
0,0254	0,7829	0,6661	-0,9804	0,0183	0,0251	-0,9679	0,0213	0,0223
0,0255	0,7831	0,6660	-1,0080	0,0177	0,0255	-0,9660	0,0214	0,0226
0,0254	0,7829	0,6660	-0,9809	0,0184	0,0248	-0,9930	0,0211	0,0227
0,0252	0,7828	0,6662	-0,9536	0,0191	0,0239	-0,9572	0,0221	0,0221
0,0252	0,7826	0,6664	-0,9821	0,0179	0,0233	-0,9828	0,0233	0,0233
0,0252	0,7825	0,6661	-1,0078	0,0172	0,0237	-0,9853	0,0234	0,0239
0,0253	0,7828	0,6657	-0,9807	0,0169	0,0227	-0,9589	0,0240	0,0237
0,0251	0,7830	0,6658	-0,9532	0,0175	0,0224	-0,9936	0,0239	0,0238
0,0251	0,7830	0,6663	-0,9821	0,0155	0,0220	-0,9699	0,0246	0,0238
0,0255	0,7831	0,6662	-1,0072	0,0149	0,0227	-0,9688	0,0249	0,0243
0,0256	0,7829	0,6658	-0,9777	0,0151	0,0225	-0,9955	0,0246	0,0246
0,0257	0,7831	0,6656	-0,9528	0,0158	0,0224	-0,9595	0,0253	0,0242

0,0258	0,7829	0,6655	-0,9833	0,0160	0,0221	-0,9836	0,0227	0,0217
0,0258	0,7830	0,6654	-1,0075	0,0159	0,0232	-0,9853	0,0228	0,0221
0,0255	0,7830	0,6655	-0,9774	0,0169	0,0226	-0,9580	0,0234	0,0217
0,0255	0,7831	0,6656	-0,9516	0,0178	0,0222	-0,9927	0,0231	0,0228
0,0257	0,7829	0,6658	-0,9827	0,0181	0,0230	-0,9695	0,0236	0,0228
0,0257	0,7827	0,6659	-1,0073	0,0178	0,0236	-0,9673	0,0238	0,0231
0,0257	0,7828	0,6658	-0,9774	0,0186	0,0232	-0,9916	0,0234	0,0237
0,0255	0,7829	0,6658	-0,9512	0,0192	0,0230	-0,9556	0,0241	0,0237
0,0254	0,7828	0,6657	-0,9817	0,0182	0,0227	-0,9819	0,0256	0,0261
0,0256	0,7828	0,6655	-1,0050	0,0179	0,0232	-0,9815	0,0255	0,0261
0,0255	0,7826	0,6656	-0,9731	0,0185	0,0225	-0,9580	0,0258	0,0257
0,0258	0,7828	0,6657	-0,9483	0,0193	0,0223	-0,9928	0,0257	0,0267
0,0254	0,7827	0,6661	-0,9807	0,0205	0,0236	-0,9662	0,0263	0,0268
0,0254	0,7826	0,6659	-1,0030	0,0206	0,0243	-0,9675	0,0262	0,0267
0,0252	0,7826	0,6658	-0,9725	0,0224	0,0241	-0,9904	0,0259	0,0269
0,0251	0,7826	0,6663	-0,9474	0,0229	0,0230	-0,9550	0,0267	0,0266
0,0250	0,7825	0,6668	-0,9796	0,0246	0,0240	-0,9808	0,0287	0,0281
0,0250	0,7823	0,6667	-1,0023	0,0242	0,0248	-0,9804	0,0282	0,0279
0,0249	0,7822	0,6666	-0,9722	0,0254	0,0246	-0,9570	0,0286	0,0276
0,0250	0,7822	0,6665	-0,9487	0,0258	0,0247	-0,9918	0,0286	0,0286
0,0247	0,7823	0,6667	-0,9803	0,0263	0,0260	-0,9647	0,0288	0,0281
0,0246	0,7824	0,6669	-1,0029	0,0258	0,0266	-0,9668	0,0285	0,0280
0,0248	0,7824	0,6670	-0,9723	0,0275	0,0269	-0,9917	0,0280	0,0284
0,0248	0,7826	0,6668	-0,9485	0,0283	0,0263	-0,9559	0,0286	0,0278
0,0246	0,7825	0,6669	-0,9801	0,0283	0,0270	-0,9831	0,0282	0,0296
0,0245	0,7824	0,6666	-1,0024	0,0277	0,0271	-0,9825	0,0279	0,0294
0,0247	0,7823	0,6668	-0,9716	0,0291	0,0272	-0,9595	0,0285	0,0296
0,0245	0,7821	0,6667	-0,9481	0,0292	0,0266	-0,9942	0,0283	0,0304
0,0246	0,7821	0,6665	-0,9801	0,0301	0,0274	-0,9696	0,0287	0,0298
0,0249	0,7820	0,6663	-1,0028	0,0296	0,0277	-0,9705	0,0285	0,0293
0,0247	0,7822	0,6658	-0,9722	0,0309	0,0282	-0,9930	0,0278	0,0296
0,0244	0,7825	0,6655	-0,9475	0,0314	0,0275	-0,9571	0,0285	0,0292
0,0246	0,7824	0,6652	-0,9796	0,0317	0,0286	-0,9841	0,0297	0,0311
0,0243	0,7824	0,6651	-1,0019	0,0311	0,0290	-0,9828	0,0292	0,0311
0,0241	0,7825	0,6652	-0,9713	0,0319	0,0289	-0,9598	0,0295	0,0306
0,0242	0,7825	0,6654	-0,9482	0,0322	0,0279	-0,9945	0,0288	0,0310
0,0240	0,7827	0,6652	-0,9808	0,0325	0,0294	-0,9678	0,0293	0,0308
0,0240	0,7827	0,6654	-1,0035	0,0320	0,0299	-0,9679	0,0293	0,0305
0,0241	0,7828	0,6652	-0,9730	0,0329	0,0303	-0,9910	0,0286	0,0306
0,0243	0,7829	0,6653	-0,9492	0,0331	0,0296	-0,9555	0,0294	0,0300
0,0244	0,7830	0,6652	-0,9809	0,0340	0,0299	-0,9815	0,0308	0,0322
0,0241	0,7831	0,6651	-1,0039	0,0326	0,0301	-0,9817	0,0302	0,0320
0,0241	0,7833	0,6652	-0,9732	0,0337	0,0300	-0,9576	0,0306	0,0314
0,0240	0,7835	0,6651	-0,9496	0,0341	0,0295	-0,9923	0,0304	0,0322

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 16 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

0,0240	0,7834	0,6654	-0,9814	0,0338	0,0304	-0,9653	0,0310	0,0316
0,0240	0,7834	0,6659	-1,0040	0,0329	0,0307	-0,9663	0,0307	0,0320
0,0237	0,7834	0,6659	-0,9738	0,0337	0,0305	-0,9892	0,0300	0,0326
0,0237	0,7835	0,6657	-0,9498	0,0342	0,0296	-0,9538	0,0305	0,0319
0,0239	0,7838	0,6658	-0,9815	0,0350	0,0304	-0,9807	0,0315	0,0330
0,0241	0,7839	0,6657	-1,0039	0,0343	0,0309	-0,9795	0,0311	0,0331
0,0243	0,7836	0,6658	-0,9729	0,0349	0,0307	-0,9560	0,0315	0,0323
0,0243	0,7835	0,6658	-0,9497	0,0348	0,0302	-0,9908	0,0314	0,0327
0,0245	0,7833	0,6657	-0,9828	0,0337	0,0312	-0,9644	0,0316	0,0321
0,0245	0,7834	0,6658	-1,0031	0,0325	0,0318	-0,9656	0,0315	0,0323
0,0247	0,7833	0,6660	-0,9704	0,0329	0,0307	-0,9880	0,0310	0,0317
0,0249	0,7836	0,6660	-0,9486	0,0329	0,0302	-0,9526	0,0315	0,0315
0,0249	0,7836	0,6661	-0,9820	0,0326	0,0317	-0,9788	0,0320	0,0314
0,0247	0,7835	0,6664	-1,0024	0,0319	0,0325	-0,9777	0,0315	0,0306
0,0246	0,7834	0,6667	-0,9687	0,0322	0,0322	-0,9549	0,0318	0,0303
0,0245	0,7835	0,6668	-0,9520	0,0324	0,0320	-0,9897	0,0313	0,0312
0,0246	0,7831	0,6674	-0,9870	0,0314	0,0325	-0,9641	0,0315	0,0309
0,0246	0,7834	0,6676	-1,0051	0,0308	0,0332	-0,9667	0,0312	0,0310
0,0245	0,7836	0,6678	-0,9702	0,0311	0,0329	-0,9908	0,0305	0,0316
0,0248	0,7836	0,6681	-0,9521	0,0311	0,0318	-0,9554	0,0309	0,0309
0,0248	0,7837	0,6681	-0,9870	0,0296	0,0315	-0,9811	0,0330	0,0317
0,0250	0,7836	0,6681	-1,0048	0,0293	0,0319	-0,9801	0,0323	0,0314
0,0250	0,7834	0,6678	-0,9697	0,0297	0,0314	-0,9553	0,0327	0,0311
0,0251	0,7833	0,6680	-0,9525	0,0299	0,0308	-0,9901	0,0324	0,0316
0,0251	0,7832	0,6683	-0,9874	0,0293	0,0310	-0,9645	0,0326	0,0312
0,0251	0,7830	0,6683	-1,0054	0,0289	0,0315	-0,9663	0,0320	0,0307
0,0251	0,7830	0,6684	-0,9700	0,0293	0,0306	-0,9912	0,0312	0,0308
0,0249	0,7827	0,6682	-0,9511	0,0298	0,0300	-0,9561	0,0318	0,0304
0,0250	0,7827	0,6681	-0,9861	0,0290	0,0301	-0,9842	0,0310	0,0311
0,0250	0,7827	0,6677	-1,0039	0,0281	0,0304	-0,9785	0,0306	0,0305
0,0250	0,7827	0,6677	-0,9689	0,0288	0,0304	-0,9575	0,0310	0,0302
0,0249	0,7827	0,6678	-0,9511	0,0282	0,0293	-0,9923	0,0304	0,0310
0,0251	0,7827	0,6678	-0,9861	0,0270	0,0295	-0,9639	0,0305	0,0309
0,0252	0,7826	0,6679	-1,0038	0,0268	0,0293	-0,9702	0,0450	0,0261
0,0253	0,7828	0,6679	-0,9682	0,0275	0,0287	-0,9907	0,0439	0,0261
0,0251	0,7827	0,6682	-0,9513	0,0276	0,0285	-0,9552	0,0441	0,0256
0,0251	0,7825	0,6682	-0,9863	0,0268	0,0289	-0,9848	0,0415	0,0255
0,0250	0,7824	0,6681	-1,0045	0,0260	0,0293	-0,9805	0,0407	0,0251
0,0251	0,7824	0,6680	-0,9695	0,0267	0,0285	-0,9602	0,0407	0,0246
0,0250	0,7824	0,6680	-0,9510	0,0270	0,0279	-0,9948	0,0397	0,0253
0,0252	0,7825	0,6679	-0,9860	0,0262	0,0281	-0,9656	0,0398	0,0248
0,0252	0,7827	0,6675	-1,0037	0,0259	0,0282	-0,9614	0,0533	0,0393
0,0253	0,7829	0,6672	-0,9690	0,0264	0,0276	-0,9831	0,0519	0,0393
0,0253	0,7830	0,6672	-0,9513	0,0266	0,0270	-0,9482	0,0520	0,0384

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 17 dari 58

0,0255	0,7828	0,6672	-0,9863	0,0259	0,0272	-0,9763	0,0508	0,0389
0,0254	0,7828	0,6674	-1,0021	0,0261	0,0273	-0,9715	0,0499	0,0383
0,0256	0,7829	0,6672	-0,9656	0,0269	0,0267	-0,9492	0,0496	0,0378
0,0257	0,7829	0,6673	-0,9504	0,0271	0,0268	-0,9842	0,0482	0,0377
0,0257	0,7829	0,6671	-0,9854	0,0270	0,0273	-0,9566	0,0479	0,0370
0,0258	0,7829	0,6673	-1,0019	0,0269	0,0276	-0,9699	0,0483	0,0185
0,0256	0,7829	0,6674	-0,9659	0,0271	0,0272	-0,9919	0,0472	0,0187
0,0256	0,7829	0,6669	-0,9512	0,0273	0,0270	-0,9569	0,0471	0,0187
0,0253	0,7828	0,6668	-0,9861	0,0268	0,0269	-0,9870	0,0437	0,0184
0,0253	0,7827	0,6668	-1,0026	0,0268	0,0274	-0,9783	0,0432	0,0191
0,0253	0,7827	0,6667	-0,9663	0,0266	0,0263	-0,9605	0,0430	0,0189
0,0255	0,7826	0,6671	-0,9500	0,0273	0,0267	-0,9951	0,0418	0,0189
0,0255	0,7827	0,6668	-0,9850	0,0270	0,0270	-0,9644	0,0414	0,0189
0,0254	0,7827	0,6668	-1,0019	0,0272	0,0275	-0,9567	0,0362	0,0392
0,0254	0,7826	0,6667	-0,9664	0,0277	0,0269	-0,9752	0,0354	0,0390
0,0253	0,7825	0,6671	-0,9522	0,0270	0,0262	-0,9410	0,0356	0,0386
0,0253	0,7825	0,6672	-0,9871	0,0268	0,0271	-0,9715	0,0339	0,0373
0,0254	0,7824	0,6669	-0,9999	0,0263	0,0278	-0,9637	0,0336	0,0373
0,0254	0,7827	0,6670	-0,9630	0,0273	0,0275	-0,9463	0,0338	0,0371
0,0254	0,7827	0,6673	-0,9493	0,0273	0,0265	-0,9813	0,0330	0,0372
0,0254	0,7828	0,6674	-0,9843	0,0272	0,0272	-0,9502	0,0332	0,0366
0,0254	0,7827	0,6672	-0,9978	0,0269	0,0273	-0,9666	0,0335	0,0296
0,0256	0,7825	0,6668	-0,9614	0,0275	0,0267	-0,9857	0,0326	0,0294
0,0256	0,7829	0,6667	-0,9487	0,0270	0,0258	-0,9518	0,0328	0,0288
0,0256	0,7829	0,6665	-0,9837	0,0263	0,0271	-0,9826	0,0304	0,0278
0,0256	0,7830	0,6664	-0,9968	0,0260	0,0280	-0,9746	0,0303	0,0279
0,0254	0,7830	0,6665	-0,9602	0,0269	0,0272	-0,9547	0,0306	0,0274
0,0252	0,7831	0,6662	-0,9474	0,0271	0,0269	-0,9889	0,0301	0,0276
0,0254	0,7832	0,6661	-0,9825	0,0267	0,0279	-0,9587	0,0304	0,0268
0,0255	0,7830	0,6660	-0,9961	0,0266	0,0284	-0,9760	0,0298	0,0185
0,0254	0,7830	0,6663	-0,9593	0,0272	0,0280	-0,9954	0,0290	0,0187
0,0250	0,7829	0,6662	-0,9451	0,0271	0,0272	-0,9608	0,0296	0,0187
0,0249	0,7830	0,6665	-0,9803	0,0261	0,0282	-0,9905	0,0275	0,0185
0,0248	0,7827	0,6665	-0,9935	0,0257	0,0289	-0,9785	0,0275	0,0189
0,0248	0,7828	0,6666	-0,9568	0,0268	0,0282	-0,9633	0,0273	0,0189
0,0247	0,7830	0,6662	-0,9429	0,0269	0,0281	-0,9967	0,0264	0,0193
0,0248	0,7828	0,6663	-0,9781	0,0267	0,0290	-0,9647	0,0271	0,0193
0,0249	0,7831	0,6664	-0,9913	0,0261	0,0297	-0,9790	0,0314	0,0179
0,0249	0,7830	0,6665	-0,9545	0,0269	0,0291	-0,9948	0,0310	0,0181
0,0249	0,7830	0,6665	-0,9446	0,0271	0,0290	-0,9616	0,0314	0,0184
0,0247	0,7831	0,6666	-0,9798	0,0262	0,0299	-0,9939	0,0308	0,0185
0,0248	0,7829	0,6665	-0,9930	0,0256	0,0279	-0,9841	0,0308	0,0189
0,0249	0,7832	0,6665	-0,9561	0,0265	0,0277	-0,9677	0,0308	0,0189
0,0249	0,7830	0,6667	-0,9452	0,0268	0,0270	-1,0000	0,0296	0,0195

0,0246	0,7831	0,6669	-0,9804	0,0264	0,0278	-0,9672	0,0301	0,0193
0,0247	0,7832	0,6673	-0,9906	0,0253	0,0283	-0,9815	0,0287	0,0126
0,0248	0,7832	0,6671	-0,9535	0,0262	0,0276	-0,9958	0,0283	0,0133
0,0250	0,7835	0,6669	-0,9433	0,0263	0,0271	-0,9616	0,0289	0,0138
0,0251	0,7835	0,6665	-0,9786	0,0251	0,0277	-0,9940	0,0290	0,0149
0,0252	0,7834	0,6667	-0,9898	0,0242	0,0275	-0,9836	0,0289	0,0155
0,0250	0,7831	0,6667	-0,9530	0,0252	0,0266	-0,9680	0,0292	0,0158
0,0250	0,7832	0,6668	-0,9434	0,0249	0,0265	-1,0009	0,0285	0,0167
0,0252	0,7831	0,6671	-0,9786	0,0236	0,0273	-0,9681	0,0288	0,0168
0,0252	0,7831	0,6670	-0,9869	0,0234	0,0283	-0,9838	0,0303	0,0096
0,0253	0,7831	0,6667	-0,9496	0,0242	0,0275	-0,9960	0,0298	0,0103
0,0252	0,7831	0,6667	-0,9416	0,0246	0,0259	-0,9632	0,0304	0,0106
0,0252	0,7829	0,6666	-0,9769	0,0239	0,0268	-0,9965	0,0310	0,0117
0,0254	0,7831	0,6664	-0,9868	0,0236	0,0275	-0,9812	0,0312	0,0121
0,0251	0,7830	0,6662	-0,9491	0,0245	0,0275	-0,9673	0,0311	0,0123

2.1.2.2.3 Tes Sumbu Y pada 1,0 G

Acc X	Acc Y	Acc Z	10 Hz			15 Hz			20 Hz		
			Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9
0,1313	0,9971	0,0201	0,1228	0,9972	0,0222	0,1232	1,0143	0,0220			
0,1283	0,9599	0,0198	0,1244	1,0216	0,0223	0,1223	1,0063	0,0221			
0,1284	0,9667	0,0197	0,1217	0,9868	0,0224	0,1230	1,0154	0,0220			
0,1310	1,0011	0,0202	0,1236	1,0120	0,0226	0,1222	1,0060	0,0222			
0,1297	0,9964	0,0186	0,1238	1,0138	0,0227	0,1229	1,0143	0,0224			
0,1267	0,9594	0,0182	0,1220	0,9898	0,0222	0,1225	1,0076	0,0227			
0,1271	0,9667	0,0183	0,1244	1,0235	0,0227	0,1230	1,0152	0,0225			
0,1297	1,0011	0,0188	0,1227	1,0005	0,0226	0,1226	1,0061	0,0225			
0,1281	0,9965	0,0175	0,1228	1,0001	0,0225	0,1232	1,0160	0,0224			
0,1251	0,9596	0,0175	0,1245	1,0257	0,0230	0,1226	1,0072	0,0228			
0,1251	0,9653	0,0177	0,1221	0,9906	0,0227	0,1230	1,0142	0,0225			
0,1280	0,9997	0,0182	0,1239	1,0154	0,0231	0,1224	1,0061	0,0224			
0,1294	0,9972	0,0171	0,1239	1,0173	0,0229	0,1231	1,0149	0,0222			
0,1266	0,9602	0,0163	0,1223	0,9927	0,0224	0,1225	1,0080	0,0223			
0,1267	0,9658	0,0164	0,1249	1,0262	0,0227	0,1229	1,0139	0,0221			
0,1296	1,0003	0,0171	0,1232	1,0021	0,0222	0,1226	1,0094	0,0217			
0,1293	0,9961	0,0155	0,1229	1,0000	0,0222	0,1230	1,0157	0,0217			
0,1261	0,9594	0,0152	0,1245	1,0232	0,0225	0,1224	1,0099	0,0215			
0,1258	0,9657	0,0157	0,1222	0,9883	0,0227	0,1224	1,0131	0,0218			
0,1287	1,0002	0,0162	0,1238	1,0123	0,0230	0,1223	1,0120	0,0221			
0,1274	0,9948	0,0153	0,1240	1,0130	0,0231	0,1225	1,0137	0,0224			

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 19 dari 58

0,1247	0,9580	0,0151	0,1222	0,9872	0,0230	0,1225	1,0133	0,0222
0,1247	0,9638	0,0156	0,1248	1,0209	0,0235	0,1226	1,0156	0,0221
0,1276	0,9983	0,0166	0,1228	0,9961	0,0228	0,1227	1,0125	0,0220
0,1282	0,9949	0,0155	0,1228	0,9958	0,0227	0,1228	1,0139	0,0214
0,1255	0,9584	0,0151	0,1248	1,0218	0,0228	0,1230	1,0120	0,0214
0,1258	0,9667	0,0157	0,1222	0,9871	0,0222	0,1232	1,0145	0,0218
0,1282	1,0011	0,0164	0,1241	1,0125	0,0224	0,1229	1,0132	0,0219
0,1255	0,9959	0,0161	0,1240	1,0125	0,0230	0,1230	1,0160	0,0220
0,1227	0,9593	0,0160	0,1221	0,9881	0,0231	0,1230	1,0145	0,0222
0,1229	0,9672	0,0160	0,1246	1,0219	0,0234	0,1232	1,0174	0,0226
0,1258	1,0016	0,0168	0,1228	0,9974	0,0235	0,1234	1,0174	0,0225
0,1230	0,9950	0,0172	0,1228	0,9980	0,0230	0,1233	1,0146	0,0226
0,1201	0,9583	0,0168	0,1245	1,0240	0,0229	0,1232	1,0153	0,0226
0,1205	0,9663	0,0174	0,1219	0,9892	0,0230	0,1232	1,0128	0,0227
0,1234	1,0008	0,0179	0,1239	1,0153	0,0232	0,1234	1,0131	0,0226
0,1221	0,9935	0,0193	0,1238	1,0125	0,0232	0,1236	1,0111	0,0227
0,1195	0,9570	0,0188	0,1221	0,9892	0,0228	0,1237	1,0137	0,0231
0,1204	0,9671	0,0200	0,1247	1,0229	0,0230	0,1232	1,0078	0,0233
0,1231	1,0016	0,0209	0,1226	0,9947	0,0231	0,1235	1,0120	0,0234
0,1226	0,9937	0,0225	0,1228	1,0006	0,0232	0,1231	1,0058	0,0231
0,1199	0,9571	0,0218	0,1228	1,0186	0,0229	0,1235	1,0100	0,0226
0,1202	0,9650	0,0223	0,1206	0,9845	0,0230	0,1230	1,0044	0,0222
0,1227	0,9995	0,0228	0,1227	1,0127	0,0234	0,1234	1,0090	0,0225
0,1207	0,9943	0,0227	0,1222	1,0065	0,0233	0,1230	1,0054	0,0225
0,1181	0,9581	0,0222	0,1205	0,9854	0,0230	0,1238	1,0113	0,0224
0,1191	0,9681	0,0235	0,1231	1,0192	0,0231	0,1236	1,0069	0,0225
0,1220	1,0025	0,0245	0,1211	0,9905	0,0230	0,1241	1,0127	0,0227
0,1200	0,9952	0,0251	0,1214	0,9959	0,0227	0,1240	1,0083	0,0232
0,1172	0,9588	0,0246	0,1223	1,0155	0,0228	0,1243	1,0115	0,0230
0,1177	0,9693	0,0253	0,1199	0,9813	0,0224	0,1239	1,0059	0,0230
0,1205	1,0036	0,0252	0,1220	1,0097	0,0226	0,1243	1,0105	0,0231
0,1183	0,9972	0,0251	0,1214	1,0030	0,0226	0,1240	1,0044	0,0231
0,1159	0,9612	0,0249	0,1200	0,9835	0,0223	0,1244	1,0089	0,0231
0,1163	0,9693	0,0249	0,1226	1,0171	0,0224	0,1238	1,0030	0,0235
0,1194	1,0037	0,0253	0,1209	0,9892	0,0221	0,1243	1,0092	0,0239
0,1169	0,9979	0,0249	0,1217	0,9979	0,0218	0,1240	1,0041	0,0236
0,1142	0,9617	0,0244	0,1229	1,0152	0,0220	0,1243	1,0100	0,0233
0,1150	0,9718	0,0257	0,1205	0,9817	0,0218	0,1240	1,0064	0,0236
0,1179	1,0061	0,0260	0,1227	1,0116	0,0219	0,1241	1,0122	0,0235
0,1167	0,9988	0,0270	0,1219	1,0020	0,0220	0,1239	1,0076	0,0230
0,1142	0,9622	0,0262	0,1206	0,9847	0,0219	0,1241	1,0107	0,0229
0,1148	0,9726	0,0272	0,1232	1,0176	0,0222	0,1241	1,0054	0,0228
0,1180	1,0069	0,0278	0,1212	0,9873	0,0218	0,1241	1,0090	0,0231
0,1158	1,0000	0,0276	0,1220	0,9966	0,0220	0,1233	1,0042	0,0235

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 20 dari 58

0,1134	0,9637	0,0269	0,1221	1,0154	0,0216	0,1236	1,0102	0,0235
0,1141	0,9723	0,0275	0,1197	0,9822	0,0215	0,1231	1,0053	0,0235
0,1171	1,0066	0,0277	0,1221	1,0133	0,0217	0,1233	1,0102	0,0235
0,1149	0,9994	0,0279	0,1215	1,0045	0,0218	0,1228	1,0036	0,0235
0,1124	0,9630	0,0275	0,1203	0,9888	0,0218	0,1231	1,0094	0,0232
0,1130	0,9703	0,0270	0,1229	1,0219	0,0218	0,1227	1,0061	0,0234
0,1161	1,0046	0,0270	0,1210	0,9913	0,0214	0,1230	1,0115	0,0233
0,1146	0,9996	0,0262	0,1218	0,9994	0,0213	0,1230	1,0083	0,0234
0,1120	0,9629	0,0256	0,1233	1,0170	0,0220	0,1232	1,0141	0,0234
0,1129	0,9736	0,0261	0,1208	0,9833	0,0219	0,1231	1,0093	0,0230
0,1159	1,0078	0,0269	0,1230	1,0141	0,0225	0,1234	1,0130	0,0230
0,1137	1,0010	0,0270	0,1223	1,0050	0,0227	0,1231	1,0075	0,0227
0,1112	0,9647	0,0264	0,1210	0,9873	0,0225	0,1236	1,0114	0,0225
0,1122	0,9735	0,0271	0,1232	1,0205	0,0225	0,1231	1,0060	0,0224
0,1155	1,0077	0,0277	0,1209	0,9910	0,0224	0,1235	1,0104	0,0225
0,1131	1,0010	0,0276	0,1214	1,0008	0,0229	0,1229	1,0044	0,0221
0,1109	0,9642	0,0273	0,1229	1,0181	0,0237	0,1233	1,0086	0,0220
0,1120	0,9741	0,0285	0,1205	0,9852	0,0234	0,1229	1,0022	0,0221
0,1153	1,0083	0,0288	0,1227	1,0162	0,0234	0,1236	1,0116	0,0223
0,1144	1,0011	0,0302	0,1220	1,0079	0,0233	0,1229	1,0043	0,0221
0,1121	0,9643	0,0296	0,1209	0,9915	0,0232	0,1234	1,0119	0,0220
0,1133	0,9741	0,0303	0,1236	1,0249	0,0236	0,1227	1,0030	0,0225
0,1166	1,0083	0,0306	0,1216	0,9940	0,0231	0,1231	1,0102	0,0226
0,1161	1,0008	0,0315	0,1223	1,0039	0,0232	0,1222	1,0018	0,0225
0,1136	0,9642	0,0311	0,1240	1,0210	0,0242	0,1229	1,0095	0,0225
0,1143	0,9745	0,0323	0,1219	0,9879	0,0243	0,1220	0,9995	0,0224
0,1172	1,0087	0,0326	0,1241	1,0185	0,0249	0,1229	1,0090	0,0225
0,1185	0,9980	0,0328	0,1231	1,0079	0,0246	0,1224	1,0011	0,0224
0,1160	0,9621	0,0320	0,1221	0,9906	0,0243	0,1231	1,0114	0,0224
0,1169	0,9737	0,0310	0,1244	1,0240	0,0245	0,1225	1,0036	0,0223
0,1196	1,0079	0,0312	0,1222	0,9932	0,0247	0,1233	1,0133	0,0224
0,1191	0,9969	0,0305	0,1228	1,0060	0,0248	0,1229	1,0039	0,0223
0,1166	0,9620	0,0296	0,1234	1,0220	0,0244	0,1236	1,0119	0,0224
0,1172	0,9761	0,0284	0,1209	0,9897	0,0243	0,1230	1,0028	0,0224
0,1199	1,0102	0,0284	0,1234	1,0214	0,0247	0,1236	1,0103	0,0229
0,1192	0,9991	0,0277	0,1223	1,0077	0,0244	0,1229	1,0006	0,0231
0,1166	0,9638	0,0273	0,1211	0,9938	0,0244	0,1235	1,0090	0,0226
0,1182	0,9811	0,0275	0,1235	1,0258	0,0248	0,1230	1,0021	0,0229
0,1210	1,0151	0,0276	0,1211	0,9940	0,0245	0,1240	1,0126	0,0230
0,1184	1,0008	0,0271	0,1218	1,0050	0,0245	0,1231	1,0006	0,0232
0,1162	0,9661	0,0264	0,1230	1,0187	0,0248	0,1239	1,0108	0,0230
0,1181	0,9828	0,0269	0,1207	0,9866	0,0245	0,1229	0,9993	0,0229
0,1208	1,0167	0,0269	0,1232	1,0191	0,0247	0,1236	1,0137	0,0230
0,1179	1,0035	0,0259	0,1225	1,0088	0,0246	0,1224	0,9972	0,0223

0,1154	0,9685	0,0251	0,1214	0,9943	0,0242	0,1235	1,0134	0,0227
0,1167	0,9864	0,0253	0,1236	1,0266	0,0242	0,1225	0,9994	0,0226
0,1194	1,0202	0,0254	0,1215	0,9944	0,0240	0,1235	1,0155	0,0229
0,1164	1,0062	0,0252	0,1222	1,0068	0,0238	0,1226	1,0007	0,0231
0,1140	0,9715	0,0246	0,1234	1,0206	0,0238	0,1237	1,0168	0,0232
0,1160	0,9878	0,0247	0,1210	0,9879	0,0241	0,1229	1,0009	0,0230
0,1188	1,0216	0,0251	0,1233	1,0205	0,0241	0,1241	1,0170	0,0232
0,1161	1,0068	0,0247	0,1227	1,0094	0,0241	0,1231	1,0032	0,0231
0,1137	0,9716	0,0242	0,1216	0,9963	0,0238	0,1242	1,0188	0,0231
0,1151	0,9877	0,0244	0,1237	1,0281	0,0240	0,1229	1,0048	0,0232
0,1177	1,0214	0,0246	0,1215	0,9955	0,0239	0,1241	1,0206	0,0235
0,1156	1,0086	0,0238	0,1223	1,0091	0,0242	0,1229	1,0052	0,0229
0,1132	0,9733	0,0236	0,1229	1,0239	0,0241	0,1241	1,0197	0,0233
0,1148	0,9896	0,0239	0,1204	0,9917	0,0240	0,1228	1,0039	0,0237
0,1178	1,0233	0,0246	0,1230	1,0229	0,0240	0,1239	1,0183	0,0239
0,1156	1,0083	0,0244	0,1221	1,0096	0,0238	0,1227	1,0022	0,0239
0,1133	0,9731	0,0235	0,1210	0,9956	0,0233	0,1237	1,0165	0,0236
0,1152	0,9909	0,0240	0,1236	1,0279	0,0231	0,1223	0,9996	0,0230
0,1178	1,0245	0,0241	0,1212	0,9962	0,0231	0,1235	1,0161	0,0227
0,1155	1,0108	0,0236	0,1218	1,0070	0,0233	0,1224	1,0016	0,0226
0,1131	0,9765	0,0230	0,1232	1,0209	0,0231	0,1237	1,0178	0,0225
0,1148	0,9935	0,0232	0,1209	0,9881	0,0232	0,1227	1,0034	0,0224
0,1176	1,0271	0,0237	0,1233	1,0208	0,0238	0,1239	1,0193	0,0222
0,1161	1,0116	0,0239	0,1224	1,0091	0,0236	0,1226	1,0043	0,0224
0,1138	0,9763	0,0236	0,1213	0,9957	0,0237	0,1238	1,0209	0,0226
0,1153	0,9923	0,0239	0,1236	1,0281	0,0238	0,1228	1,0058	0,0228
0,1177	1,0258	0,0243	0,1215	0,9961	0,0235	0,1238	1,0217	0,0229
0,1154	1,0105	0,0247	0,1226	1,0099	0,0235	0,1231	1,0068	0,0230
0,1132	0,9751	0,0237	0,1233	1,0238	0,0234	0,1245	1,0238	0,0228
0,1147	0,9922	0,0241	0,1212	0,9922	0,0234	0,1232	1,0075	0,0228
0,1178	1,0258	0,0246	0,1236	1,0235	0,0236	0,1243	1,0251	0,0229
0,1161	1,0103	0,0245	0,1227	1,0100	0,0239	0,1232	1,0077	0,0228
0,1138	0,9754	0,0239	0,1219	0,9958	0,0236	0,1243	1,0226	0,0227
0,1153	0,9907	0,0240	0,1241	1,0280	0,0233	0,1230	1,0049	0,0221
0,1178	1,0244	0,0246	0,1218	0,9967	0,0232	0,1244	1,0255	0,0221
0,1152	1,0101	0,0247	0,1226	1,0089	0,0232	0,1230	1,0048	0,0216
0,1129	0,9754	0,0247	0,1239	1,0221	0,0236	0,1246	1,0238	0,0214
0,1143	0,9932	0,0255	0,1217	0,9899	0,0238	0,1234	1,0055	0,0211
0,1172	1,0267	0,0256	0,1238	1,0217	0,0237	0,1247	1,0249	0,0218
0,1149	1,0135	0,0247	0,1233	1,0106	0,0240	0,1234	1,0067	0,0217
0,1125	0,9786	0,0238	0,1222	0,9964	0,0242	0,1248	1,0260	0,0221
0,1141	0,9940	0,0239	0,1246	1,0287	0,0242	0,1231	1,0070	0,0220
0,1172	1,0275	0,0245	0,1220	0,9963	0,0239	0,1245	1,0279	0,0223
0,1180	1,0106	0,0251	0,1229	1,0077	0,0238	0,1230	1,0068	0,0222

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 22 dari 58

0,1156	0,9766	0,0247	0,1238	1,0229	0,0242	0,1244	1,0257	0,0224
0,1170	0,9965	0,0248	0,1216	0,9910	0,0239	0,1231	1,0056	0,0223
0,1197	1,0293	0,0250	0,1239	1,0227	0,0235	0,1244	1,0236	0,0220
0,1196	1,0131	0,0258	0,1227	1,0096	0,0233	0,1231	1,0031	0,0218
0,1173	0,9793	0,0248	0,1217	0,9945	0,0233	0,1244	1,0225	0,0224
0,1187	0,9980	0,0252	0,1240	1,0266	0,0232	0,1229	1,0035	0,0226
0,1213	1,0307	0,0254	0,1217	0,9942	0,0233	0,1244	1,0235	0,0224
0,1214	1,0142	0,0262	0,1224	1,0065	0,0240	0,1233	1,0048	0,0226
0,1187	0,9807	0,0255	0,1238	1,0199	0,0244	0,1247	1,0250	0,0232
0,1204	1,0021	0,0252	0,1214	0,9874	0,0242	0,1233	1,0061	0,0234
0,1227	1,0336	0,0254	0,1238	1,0203	0,0241	0,1251	1,0265	0,0238
0,1204	1,0111	0,0255	0,1230	1,0078	0,0243	0,1235	1,0070	0,0233
0,1180	0,9790	0,0253	0,1219	0,9929	0,0243	0,1250	1,0275	0,0234
0,1197	1,0020	0,0250	0,1243	1,0259	0,0242	0,1234	1,0077	0,0230
0,1221	1,0325	0,0259	0,1219	0,9939	0,0235	0,1249	1,0286	0,0233
0,1199	1,0122	0,0257	0,1224	1,0053	0,0233	0,1231	1,0073	0,0230
0,1176	0,9803	0,0255	0,1235	1,0189	0,0232	0,1247	1,0268	0,0233
0,1194	1,0041	0,0257	0,1213	0,9864	0,0234	0,1232	1,0057	0,0230
0,1220	1,0347	0,0265	0,1236	1,0192	0,0233	0,1245	1,0249	0,0232
0,1196	1,0140	0,0261	0,1227	1,0058	0,0231	0,1230	1,0042	0,0230
0,1174	0,9821	0,0256	0,1215	0,9908	0,0231	0,1246	1,0232	0,0233
0,1191	1,0048	0,0252	0,1241	1,0225	0,0236	0,1231	1,0043	0,0226
0,1218	1,0351	0,0253	0,1219	0,9909	0,0235	0,1247	1,0240	0,0229
0,1195	1,0146	0,0249	0,1227	1,0039	0,0234	0,1232	1,0023	0,0229
0,1171	0,9826	0,0246	0,1240	1,0175	0,0230	0,1246	1,0218	0,0231
0,1188	1,0062	0,0244	0,1217	0,9852	0,0229	0,1228	0,9997	0,0231
0,1212	1,0375	0,0250	0,1240	1,0164	0,0232	0,1245	1,0208	0,0231
0,1194	1,0152	0,0248	0,1231	1,0043	0,0227	0,1232	1,0011	0,0230
0,1173	0,9821	0,0245	0,1222	0,9916	0,0229	0,1247	1,0221	0,0231
0,1195	1,0057	0,0245	0,1247	1,0239	0,0232	0,1231	1,0027	0,0228
0,1220	1,0367	0,0251	0,1225	0,9924	0,0230	0,1247	1,0231	0,0229
0,1198	1,0142	0,0249	0,1234	1,0057	0,0228	0,1234	1,0032	0,0231
0,1175	0,9833	0,0248	0,1242	1,0202	0,0228	0,1248	1,0253	0,0235
0,1195	1,0094	0,0250	0,1219	0,9873	0,0225	0,1229	1,0040	0,0233
0,1216	1,0386	0,0253	0,1242	1,0193	0,0229	0,1243	1,0233	0,0239
0,1204	1,0140	0,0256	0,1236	1,0079	0,0228	0,1228	1,0024	0,0236
0,1180	0,9825	0,0248	0,1226	0,9941	0,0229	0,1245	1,0238	0,0240
0,1199	1,0083	0,0252	0,1252	1,0263	0,0236	0,1230	1,0030	0,0238
0,1224	1,0382	0,0256	0,1227	0,9946	0,0235	0,1247	1,0227	0,0237
0,1218	1,0139	0,0256	0,1235	1,0083	0,0233	0,1230	1,0004	0,0242
0,1196	0,9822	0,0253	0,1240	1,0229	0,0233	0,1246	1,0241	0,0248
0,1216	1,0082	0,0254	0,1218	0,9911	0,0233	0,1229	0,9993	0,0250
0,1237	1,0378	0,0258	0,1240	1,0228	0,0234	0,1246	1,0225	0,0253
0,1224	1,0134	0,0253	0,1235	1,0112	0,0235	0,1227	0,9978	0,0246

0,1202	0,9819	0,0245	0,1225	0,9980	0,0232	0,1244	1,0222	0,0248
0,1223	1,0077	0,0247	0,1248	1,0300	0,0236	0,1227	0,9978	0,0243
0,1246	1,0374	0,0251	0,1227	0,9976	0,0237	0,1245	1,0207	0,0244

2.1.2.2.4 Tes Sumbu Y pada 1,5 G

10 Hz			15 Hz			20 Hz		
Acc X	Acc Y	Acc Z	Acc X2	Acc Y3	Acc Z4	Acc X3	Acc Y4	Acc Z5
0,1273	0,8738	0,0203	0,1253	0,9234	0,0245	0,1244	1,0145	0,0234
0,1299	0,8829	0,0203	0,1212	0,8724	0,0244	0,1236	1,0049	0,0231
0,1257	0,8330	0,0201	0,1246	0,9100	0,0246	0,1242	1,0134	0,0231
0,1255	0,8289	0,0190	0,1230	0,9105	0,0238	0,1239	1,0073	0,0233
0,1294	0,8679	0,0193	0,1201	0,8762	0,0235	0,1245	1,0166	0,0231
0,1309	0,8756	0,0207	0,1242	0,9137	0,0240	0,1238	1,0044	0,0226
0,1268	0,8260	0,0205	0,1217	0,8782	0,0239	0,1244	1,0134	0,0221
0,1260	0,8196	0,0203	0,1216	0,8817	0,0238	0,1236	1,0017	0,0224
0,1298	0,8589	0,0210	0,1242	0,9190	0,0241	0,1243	1,0165	0,0225
0,1299	0,8784	0,0202	0,1202	0,8684	0,0231	0,1232	1,0021	0,0226
0,1254	0,8288	0,0198	0,1237	0,9061	0,0233	0,1243	1,0196	0,0235
0,1254	0,8222	0,0189	0,1245	0,9112	0,0235	0,1233	1,0055	0,0230
0,1295	0,8615	0,0196	0,1216	0,8767	0,0235	0,1240	1,0210	0,0231
0,1300	0,8832	0,0179	0,1257	0,9141	0,0240	0,1226	1,0011	0,0232
0,1258	0,8330	0,0174	0,1228	0,8787	0,0237	0,1248	1,0306	0,0234
0,1254	0,8265	0,0167	0,1227	0,8857	0,0238	0,1226	1,0011	0,0234
0,1296	0,8656	0,0175	0,1254	0,9229	0,0241	0,1249	1,0277	0,0238
0,1371	0,8843	0,0188	0,1216	0,8725	0,0242	0,1229	1,0000	0,0236
0,1330	0,8346	0,0178	0,1250	0,9101	0,0242	0,1249	1,0249	0,0239
0,1313	0,8368	0,0181	0,1233	0,9130	0,0232	0,1226	0,9965	0,0239
0,1351	0,8756	0,0189	0,1208	0,8813	0,0233	0,1249	1,0225	0,0242
0,1327	0,8768	0,0226	0,1251	0,9186	0,0236	0,1231	0,9953	0,0241
0,1283	0,8273	0,0213	0,1226	0,8820	0,0233	0,1252	1,0252	0,0241
0,1275	0,8322	0,0209	0,1228	0,8867	0,0229	0,1233	0,9987	0,0242
0,1315	0,8711	0,0220	0,1256	0,9238	0,0231	0,1254	1,0285	0,0243
0,1313	0,8819	0,0253	0,1215	0,8732	0,0227	0,1234	1,0009	0,0239
0,1271	0,8324	0,0240	0,1248	0,9108	0,0231	0,1254	1,0254	0,0244
0,1257	0,8328	0,0250	0,1261	0,9173	0,0240	0,1235	0,9971	0,0242
0,1298	0,8717	0,0253	0,1234	0,8832	0,0239	0,1254	1,0247	0,0244
0,1283	0,8736	0,0293	0,1275	0,9205	0,0245	0,1232	0,9952	0,0242
0,1238	0,8243	0,0285	0,1248	0,8829	0,0244	0,1253	1,0223	0,0244
0,1225	0,8273	0,0287	0,1249	0,8910	0,0245	0,1232	0,9930	0,0246
0,1264	0,8664	0,0288	0,1272	0,9280	0,0248	0,1254	1,0216	0,0248
0,1282	0,8756	0,0288	0,1232	0,8776	0,0246	0,1235	0,9954	0,0245
0,1239	0,8270	0,0282	0,1265	0,9150	0,0244	0,1257	1,0259	0,0243

0,1251	0,8349	0,0278	0,1265	0,9132	0,0253	0,1238	0,9985	0,0243
0,1292	0,8738	0,0278	0,1237	0,8811	0,0248	0,1257	1,0287	0,0245
0,1320	0,8760	0,0243	0,1277	0,9184	0,0249	0,1239	1,0017	0,0244
0,1279	0,8283	0,0234	0,1246	0,8790	0,0242	0,1261	1,0314	0,0244
0,1276	0,8373	0,0233	0,1256	0,8884	0,0243	0,1237	1,0034	0,0242
0,1316	0,8760	0,0240	0,1274	0,9242	0,0250	0,1255	1,0297	0,0241
0,1324	0,8765	0,0211	0,1232	0,8737	0,0243	0,1232	1,0007	0,0236
0,1281	0,8285	0,0200	0,1267	0,9113	0,0245	0,1251	1,0272	0,0237
0,1291	0,8365	0,0195	0,1262	0,9087	0,0241	0,1230	0,9988	0,0238
0,1330	0,8753	0,0203	0,1238	0,8786	0,0239	0,1251	1,0299	0,0238
0,1357	0,8793	0,0176	0,1274	0,9159	0,0242	0,1226	0,9950	0,0235
0,1314	0,8315	0,0171	0,1250	0,8776	0,0236	0,1249	1,0285	0,0238
0,1324	0,8433	0,0183	0,1254	0,8921	0,0236	0,1224	0,9933	0,0239
0,1362	0,8818	0,0194	0,1267	0,9239	0,0238	0,1253	1,0268	0,0241
0,1395	0,8871	0,0169	0,1229	0,8743	0,0232	0,1223	0,9873	0,0239
0,1353	0,8394	0,0166	0,1258	0,9118	0,0239	0,1251	1,0210	0,0242
0,1349	0,8480	0,0163	0,1257	0,9046	0,0234	0,1223	0,9850	0,0240
0,1387	0,8864	0,0171	0,1229	0,8767	0,0232	0,1251	1,0189	0,0245
0,1392	0,8870	0,0147	0,1268	0,9141	0,0233	0,1224	0,9823	0,0241
0,1351	0,8388	0,0143	0,1231	0,8728	0,0230	0,1255	1,0162	0,0247
0,1357	0,8468	0,0139	0,1227	0,8911	0,0236	0,1226	0,9799	0,0247
0,1396	0,8852	0,0147	0,1248	0,9226	0,0239	0,1252	1,0139	0,0248
0,1425	0,8899	0,0122	0,1213	0,8730	0,0236	0,1222	0,9758	0,0244
0,1383	0,8416	0,0118	0,1247	0,9105	0,0241	0,1248	1,0100	0,0249
0,1382	0,8506	0,0121	0,1247	0,9048	0,0234	0,1219	0,9723	0,0247
0,1417	0,8889	0,0130	0,1224	0,8808	0,0232	0,1245	1,0065	0,0251
0,1438	0,8943	0,0107	0,1261	0,9181	0,0233	0,1219	0,9681	0,0247
0,1394	0,8456	0,0107	0,1223	0,8765	0,0231	0,1246	1,0025	0,0248
0,1396	0,8528	0,0101	0,1225	0,8930	0,0231	0,1218	0,9632	0,0248
0,1429	0,8911	0,0114	0,1244	0,9266	0,0232	0,1250	0,9978	0,0248
0,1444	0,8957	0,0095	0,1211	0,8779	0,0228	0,1222	0,9592	0,0246
0,1398	0,8474	0,0095	0,1248	0,9153	0,0230	0,1247	0,9939	0,0246
0,1400	0,8560	0,0084	0,1230	0,9015	0,0227	0,1218	0,9549	0,0243
0,1435	0,8941	0,0097	0,1208	0,8796	0,0227	0,1248	0,9897	0,0245
0,1453	0,8957	0,0074	0,1246	0,9169	0,0227	0,1219	0,9529	0,0242
0,1407	0,8467	0,0071	0,1209	0,8728	0,0223	0,1248	0,9879	0,0249
0,1406	0,8552	0,0065	0,1223	0,8948	0,0231	0,1222	0,9518	0,0250
0,1438	0,8934	0,0077	0,1241	0,9245	0,0230	0,1251	0,9867	0,0254
0,1461	0,8985	0,0052	0,1208	0,8773	0,0227	0,1225	0,9498	0,0253
0,1415	0,8499	0,0050	0,1247	0,9148	0,0235	0,1254	0,9848	0,0260
0,1418	0,8571	0,0047	0,1237	0,9046	0,0226	0,1222	0,9458	0,0263
0,1450	0,8952	0,0062	0,1219	0,8834	0,0222	0,1250	0,9810	0,0260
0,1471	0,9008	0,0042	0,1255	0,9207	0,0222	0,1221	0,9426	0,0258
0,1426	0,8523	0,0043	0,1220	0,8757	0,0218	0,1248	0,9778	0,0260

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 25 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

0,1426	0,8606	0,0037	0,1215	0,8944	0,0224	0,1223	0,9419	0,0261
0,1462	0,8985	0,0047	0,1232	0,9209	0,0227	0,1249	0,9772	0,0262
0,1488	0,9011	0,0021	0,1193	0,8729	0,0222	0,1222	0,9405	0,0257
0,1443	0,8527	0,0020	0,1231	0,9105	0,0224	0,1248	0,9758	0,0258
0,1441	0,8664	0,0024	0,1227	0,9007	0,0222	0,1220	0,9394	0,0253
0,1475	0,9042	0,0041	0,1210	0,8811	0,0220	0,1252	0,9748	0,0254
0,1450	0,9023	0,0033	0,1249	0,9184	0,0221	0,1223	0,9326	0,0251
0,1405	0,8548	0,0032	0,1218	0,8725	0,0223	0,1255	0,9682	0,0254
0,1403	0,8665	0,0049	0,1216	0,8917	0,0231	0,1226	0,9281	0,0254
0,1437	0,9042	0,0067	0,1234	0,9186	0,0233	0,1258	0,9639	0,0256
0,1416	0,9032	0,0055	0,1198	0,8704	0,0227	0,1230	0,9246	0,0253
0,1371	0,8555	0,0051	0,1237	0,9080	0,0229	0,1260	0,9604	0,0255
0,1373	0,8693	0,0058	0,1225	0,8965	0,0227	0,1230	0,9205	0,0252
0,1406	0,9070	0,0074	0,1212	0,8779	0,0226	0,1260	0,9565	0,0252
0,1368	0,9037	0,0074	0,1249	0,9153	0,0232	0,1226	0,9166	0,0247
0,1326	0,8550	0,0073	0,1214	0,8701	0,0228	0,1256	0,9527	0,0251
0,1323	0,8689	0,0085	0,1218	0,8907	0,0231	0,1224	0,9135	0,0246
0,1359	0,9066	0,0100	0,1236	0,9205	0,0232	0,1256	0,9497	0,0246
0,1364	0,9062	0,0074	0,1198	0,8729	0,0231	0,1227	0,9098	0,0245
0,1321	0,8582	0,0073	0,1236	0,9105	0,0233	0,1262	0,9461	0,0249
0,1327	0,8719	0,0084	0,1217	0,8975	0,0229	0,1229	0,9057	0,0245
0,1368	0,9095	0,0094	0,1197	0,8754	0,0221	0,1262	0,9422	0,0248
0,1314	0,9049	0,0102	0,1236	0,9129	0,0223	0,1230	0,9023	0,0243
0,1272	0,8569	0,0095	0,1200	0,8678	0,0222	0,1263	0,9389	0,0247
0,1283	0,8678	0,0110	0,1205	0,8884	0,0228	0,1227	0,8926	0,0243
0,1321	0,9055	0,0119	0,1221	0,9154	0,0232	0,1262	0,9295	0,0245
0,1271	0,9009	0,0134	0,1185	0,8674	0,0229	0,1230	0,8846	0,0241
0,1232	0,8533	0,0130	0,1225	0,9052	0,0233	0,1262	0,9218	0,0246
0,1235	0,8673	0,0135	0,1212	0,8833	0,0235	0,1226	0,8773	0,0244
0,1274	0,9051	0,0144	0,1202	0,8711	0,0235	0,1260	0,9147	0,0250
0,1257	0,9032	0,0129	0,1236	0,9088	0,0243	0,1225	0,8698	0,0248
0,1217	0,8556	0,0123	0,1199	0,8610	0,0236	0,1262	0,9075	0,0251
0,1223	0,8670	0,0134	0,1220	0,8892	0,0236	0,1225	0,8604	0,0249
0,1263	0,9048	0,0144	0,1230	0,9089	0,0235	0,1261	0,8983	0,0248
0,1233	0,8985	0,0183	0,1196	0,8648	0,0229	0,1225	0,8518	0,0247
0,1199	0,8517	0,0175	0,1235	0,9027	0,0231	0,1262	0,8901	0,0249
0,1202	0,8671	0,0180	0,1221	0,8837	0,0234	0,1225	0,8439	0,0243
0,1237	0,9049	0,0190	0,1207	0,8723	0,0234	0,1262	0,8825	0,0242
0,1246	0,8960	0,0226	0,1241	0,9099	0,0236	0,1228	0,8370	0,0239
0,1209	0,8511	0,0220	0,1199	0,8612	0,0229	0,1264	0,8757	0,0244
0,1230	0,8766	0,0220	0,1215	0,8931	0,0226	0,1229	0,8305	0,0242
0,1264	0,9140	0,0222	0,1228	0,9141	0,0229	0,1267	0,8694	0,0247
0,1243	0,9018	0,0193	0,1195	0,8694	0,0230	0,1230	0,8245	0,0243
0,1206	0,8568	0,0189	0,1231	0,9071	0,0229	0,1267	0,8637	0,0247

0,1228	0,8818	0,0188	0,1217	0,8842	0,0228	0,1228	0,8179	0,0242
0,1263	0,9191	0,0190	0,1209	0,8715	0,0227	0,1265	0,8573	0,0245
0,1251	0,9072	0,0165	0,1244	0,9091	0,0227	0,1228	0,8120	0,0239
0,1216	0,8622	0,0161	0,1209	0,8603	0,0223	0,1263	0,8516	0,0241
0,1241	0,8892	0,0160	0,1226	0,8922	0,0222	0,1226	0,8067	0,0237
0,1280	0,9263	0,0165	0,1234	0,9100	0,0222	0,1261	0,8464	0,0243
0,1241	0,9111	0,0148	0,1202	0,8673	0,0216	0,1225	0,8019	0,0241
0,1202	0,8656	0,0145	0,1241	0,9051	0,0218	0,1264	0,8418	0,0243
0,1225	0,8927	0,0147	0,1227	0,8826	0,0220	0,1227	0,7976	0,0237
0,1260	0,9296	0,0146	0,1220	0,8712	0,0218	0,1265	0,8377	0,0242
0,1249	0,9174	0,0128	0,1252	0,9088	0,0219	0,1232	0,7938	0,0236
0,1211	0,8715	0,0125	0,1211	0,8606	0,0215	0,1269	0,8340	0,0240
0,1239	0,8949	0,0120	0,1236	0,8900	0,0217	0,1230	0,7893	0,0235
0,1273	0,9318	0,0129	0,1246	0,9098	0,0215	0,1264	0,8296	0,0234
0,1241	0,9182	0,0108	0,1218	0,8676	0,0215	0,1227	0,7854	0,0228
0,1204	0,8722	0,0107	0,1257	0,9053	0,0220	0,1264	0,8258	0,0228
0,1234	0,8984	0,0106	0,1242	0,8838	0,0223	0,1230	0,7816	0,0226
0,1268	0,9351	0,0113	0,1232	0,8745	0,0223	0,1265	0,8223	0,0236
0,1241	0,9219	0,0088	0,1266	0,9120	0,0225	0,1228	0,7777	0,0233
0,1206	0,8763	0,0084	0,1225	0,8637	0,0217	0,1264	0,8184	0,0235
0,1237	0,9028	0,0083	0,1242	0,8939	0,0216	0,1228	0,7741	0,0229
0,1272	0,9394	0,0092	0,1246	0,9102	0,0222	0,1263	0,8150	0,0234
0,1233	0,9237	0,0077	0,1212	0,8695	0,0221	0,1225	0,7708	0,0231
0,1194	0,8774	0,0079	0,1255	0,9072	0,0225	0,1264	0,8118	0,0232
0,1224	0,9031	0,0077	0,1239	0,8825	0,0229	0,1226	0,7675	0,0229
0,1263	0,9397	0,0083	0,1229	0,8755	0,0221	0,1262	0,8086	0,0231
0,1213	0,9239	0,0070	0,1256	0,9130	0,0222	0,1227	0,7648	0,0228
0,1177	0,8779	0,0067	0,1218	0,8628	0,0221	0,1264	0,8060	0,0230
0,1206	0,9009	0,0070	0,1243	0,9007	0,0226	0,1225	0,7631	0,0226
0,1239	0,9375	0,0078	0,1267	0,9133	0,0240	0,1260	0,8043	0,0232
0,1187	0,9216	0,0075	0,1236	0,8750	0,0242	0,1224	0,7615	0,0228
0,1153	0,8756	0,0071	0,1274	0,9125	0,0247	0,1262	0,8028	0,0229
0,1178	0,8993	0,0072	0,1242	0,8812	0,0247	0,1226	0,7597	0,0228
0,1216	0,9360	0,0081	0,1236	0,8793	0,0251	0,1263	0,8011	0,0233
0,1190	0,9227	0,0061	0,1265	0,9167	0,0251	0,1228	0,7576	0,0227
0,1157	0,8770	0,0057	0,1231	0,8663	0,0250	0,1263	0,7990	0,0233
0,1180	0,9012	0,0064	0,1257	0,9036	0,0254	0,1229	0,7565	0,0230
0,1211	0,9379	0,0072	0,1253	0,9121	0,0247	0,1266	0,7980	0,0236
0,1165	0,9223	0,0056	0,1222	0,8754	0,0245	0,1228	0,7544	0,0232
0,1131	0,8765	0,0055	0,1261	0,9129	0,0248	0,1270	0,7960	0,0241
0,1163	0,9002	0,0055	0,1236	0,8781	0,0249	0,1232	0,7515	0,0237
0,1200	0,9369	0,0063	0,1236	0,8840	0,0247	0,1269	0,7931	0,0242
0,1148	0,9210	0,0058	0,1263	0,9212	0,0250	0,1226	0,7494	0,0239
0,1117	0,8752	0,0058	0,1223	0,8706	0,0245	0,1265	0,7911	0,0240

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 27 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

0,1138	0,8995	0,0065	0,1261	0,9082	0,0246	0,1225	0,7469	0,0239
0,1176	0,9362	0,0075	0,1274	0,9147	0,0243	0,1263	0,7887	0,0242
0,1148	0,9223	0,0055	0,1245	0,8786	0,0237	0,1224	0,7448	0,0240
0,1118	0,8767	0,0051	0,1283	0,9160	0,0238	0,1261	0,7867	0,0244
0,1139	0,9013	0,0058	0,1253	0,8826	0,0241	0,1222	0,7426	0,0240
0,1177	0,9379	0,0069	0,1252	0,8868	0,0243	0,1262	0,7845	0,0240
0,1144	0,9231	0,0055	0,1275	0,9239	0,0240	0,1224	0,7401	0,0241
0,1112	0,8766	0,0054	0,1236	0,8733	0,0236	0,1259	0,7821	0,0240
0,1143	0,9025	0,0052	0,1265	0,9108	0,0237	0,1223	0,7382	0,0237
0,1182	0,9391	0,0058	0,1253	0,9143	0,0235	0,1261	0,7802	0,0239
0,1136	0,9231	0,0054	0,1222	0,8775	0,0232	0,1225	0,7364	0,0238
0,1104	0,8770	0,0050	0,1263	0,9149	0,0233	0,1263	0,7785	0,0240
0,1125	0,9058	0,0054	0,1238	0,8769	0,0223	0,1225	0,7345	0,0232
0,1163	0,9423	0,0063	0,1241	0,8840	0,0225	0,1265	0,7767	0,0235
0,1138	0,9228	0,0080	0,1262	0,9212	0,0220	0,1228	0,7332	0,0235
0,1106	0,8771	0,0077	0,1224	0,8704	0,0216	0,1266	0,7754	0,0236
0,1130	0,9060	0,0086	0,1254	0,9080	0,0215	0,1229	0,7320	0,0234
0,1169	0,9425	0,0096	0,1252	0,8995	0,0211	0,1267	0,7743	0,0237
0,1151	0,9234	0,0110	0,1235	0,8789	0,0211	0,1229	0,7308	0,0235
0,1117	0,8779	0,0106	0,1274	0,9163	0,0215	0,1266	0,7731	0,0240
0,1140	0,9075	0,0112	0,1239	0,8712	0,0212	0,1230	0,7303	0,0237
0,1179	0,9439	0,0118	0,1234	0,8893	0,0222	0,1265	0,7726	0,0238
0,1196	0,9217	0,0135	0,1251	0,9181	0,0222	0,1227	0,7301	0,0233
0,1161	0,8779	0,0135	0,1213	0,8698	0,0221	0,1266	0,7725	0,0240
0,1179	0,9115	0,0147	0,1252	0,9074	0,0223	0,1228	0,7297	0,0233
0,1216	0,9478	0,0157	0,1250	0,8990	0,0217	0,1264	0,7720	0,0237
0,1219	0,9242	0,0161	0,1231	0,8772	0,0212	0,1227	0,7297	0,0234
0,1185	0,8801	0,0154	0,1271	0,9146	0,0219	0,1261	0,7721	0,0235
0,1202	0,9127	0,0164	0,1238	0,8683	0,0219	0,1222	0,7303	0,0229
0,1237	0,9489	0,0179	0,1251	0,8903	0,0224	0,1260	0,7726	0,0231
0,1237	0,9244	0,0178	0,1264	0,9147	0,0228	0,1221	0,7306	0,0228
0,1203	0,8814	0,0171	0,1228	0,8681	0,0226	0,1257	0,7730	0,0233
0,1220	0,9160	0,0183	0,1265	0,9058	0,0229	0,1222	0,7318	0,0229

2.1.2.2.5 Tes Sumbu Z pada 1,0 G

10 Hz			15 Hz			20 Hz		
X	Y	Z	Acc X	Acc Y	Acc Z	Acc X2	Acc Y3	Acc Z4
-0,0049	0,1191	1,0226	-0,0035	0,1181	1,0194	-0,0016	0,1169	1,0071
-0,0052	0,1210	1,0494	-0,0038	0,1191	1,0328	-0,0021	0,1189	1,0335
-0,0049	0,1193	1,0291	-0,0030	0,1169	1,0013	-0,0014	0,1168	1,0082
-0,0046	0,1172	1,0005	-0,0038	0,1191	1,0337	-0,0022	0,1189	1,0346
-0,0050	0,1188	1,0224	-0,0033	0,1178	1,0191	-0,0019	0,1170	1,0107
-0,0051	0,1208	1,0483	-0,0032	0,1170	1,0052	-0,0027	0,1185	1,0354

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 28 dari 58

-0,0047	0,1193	1,0279	-0,0041	0,1191	1,0364	-0,0018	0,1166	1,0091
-0,0043	0,1169	0,9997	-0,0031	0,1169	1,0036	-0,0025	0,1183	1,0328
-0,0051	0,1183	1,0219	-0,0036	0,1184	1,0158	-0,0019	0,1166	1,0072
-0,0052	0,1206	1,0486	-0,0041	0,1195	1,0283	-0,0027	0,1187	1,0313
-0,0046	0,1190	1,0271	-0,0037	0,1174	0,9975	-0,0022	0,1171	1,0055
-0,0041	0,1170	0,9981	-0,0044	0,1197	1,0309	-0,0032	0,1190	1,0290
-0,0044	0,1186	1,0212	-0,0039	0,1185	1,0151	-0,0026	0,1173	1,0039
-0,0046	0,1206	1,0479	-0,0037	0,1177	1,0039	-0,0031	0,1191	1,0303
-0,0040	0,1192	1,0264	-0,0044	0,1197	1,0340	-0,0024	0,1172	1,0054
-0,0038	0,1169	0,9980	-0,0036	0,1172	1,0010	-0,0032	0,1193	1,0313
-0,0044	0,1185	1,0207	-0,0038	0,1185	1,0187	-0,0026	0,1174	1,0055
-0,0052	0,1204	1,0480	-0,0044	0,1191	1,0299	-0,0033	0,1193	1,0292
-0,0050	0,1188	1,0267	-0,0037	0,1170	1,0002	-0,0025	0,1175	1,0046
-0,0048	0,1166	0,9989	-0,0045	0,1195	1,0330	-0,0030	0,1193	1,0305
-0,0051	0,1186	1,0218	-0,0042	0,1180	1,0170	-0,0025	0,1174	1,0055
-0,0053	0,1204	1,0488	-0,0041	0,1174	1,0081	-0,0032	0,1192	1,0317
-0,0047	0,1188	1,0273	-0,0049	0,1198	1,0387	-0,0027	0,1175	1,0067
-0,0045	0,1168	0,9989	-0,0045	0,1171	1,0049	-0,0034	0,1195	1,0302
-0,0047	0,1186	1,0216	-0,0048	0,1185	1,0225	-0,0030	0,1177	1,0058
-0,0049	0,1205	1,0466	-0,0047	0,1190	1,0312	-0,0034	0,1195	1,0309
-0,0043	0,1189	1,0241	-0,0037	0,1169	1,0002	-0,0026	0,1177	1,0064
-0,0036	0,1169	0,9968	-0,0045	0,1194	1,0335	-0,0032	0,1197	1,0320
-0,0042	0,1189	1,0206	-0,0038	0,1183	1,0183	-0,0026	0,1181	1,0073
-0,0043	0,1206	1,0464	-0,0034	0,1176	1,0084	-0,0035	0,1202	1,0335
-0,0037	0,1191	1,0242	-0,0042	0,1198	1,0387	-0,0028	0,1184	1,0095
-0,0035	0,1170	0,9970	-0,0034	0,1174	1,0060	-0,0034	0,1202	1,0333
-0,0039	0,1189	1,0205	-0,0036	0,1185	1,0230	-0,0029	0,1181	1,0068
-0,0045	0,1209	1,0458	-0,0040	0,1194	1,0340	-0,0036	0,1197	1,0315
-0,0040	0,1192	1,0232	-0,0032	0,1169	1,0035	-0,0033	0,1180	1,0079
-0,0037	0,1171	0,9967	-0,0043	0,1194	1,0367	-0,0036	0,1197	1,0329
-0,0047	0,1190	1,0221	-0,0038	0,1178	1,0190	-0,0030	0,1181	1,0107
-0,0048	0,1207	1,0458	-0,0036	0,1169	1,0088	-0,0033	0,1199	1,0321
-0,0043	0,1188	1,0219	-0,0043	0,1191	1,0391	-0,0028	0,1183	1,0095
-0,0039	0,1168	0,9964	-0,0036	0,1166	1,0047	-0,0032	0,1200	1,0311
-0,0040	0,1186	1,0224	-0,0042	0,1176	1,0210	-0,0028	0,1182	1,0082
-0,0042	0,1202	1,0471	-0,0043	0,1185	1,0307	-0,0035	0,1199	1,0303
-0,0037	0,1182	1,0220	-0,0037	0,1160	1,0002	-0,0029	0,1184	1,0098
-0,0034	0,1159	0,9960	-0,0046	0,1185	1,0335	-0,0036	0,1199	1,0324
-0,0035	0,1180	1,0217	-0,0044	0,1174	1,0187	-0,0034	0,1183	1,0113
-0,0038	0,1201	1,0461	-0,0042	0,1168	1,0087	-0,0040	0,1199	1,0341
-0,0032	0,1182	1,0225	-0,0049	0,1191	1,0379	-0,0037	0,1183	1,0127
-0,0030	0,1165	0,9972	-0,0038	0,1168	1,0050	-0,0038	0,1201	1,0338
-0,0034	0,1183	1,0234	-0,0042	0,1178	1,0223	-0,0035	0,1185	1,0117
-0,0037	0,1202	1,0479	-0,0044	0,1184	1,0323	-0,0038	0,1201	1,0324

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 29 dari 58

-0,0034	0,1183	1,0230	-0,0037	0,1159	1,0016	-0,0034	0,1184	1,0104
-0,0033	0,1164	0,9980	-0,0046	0,1181	1,0349	-0,0039	0,1202	1,0313
-0,0036	0,1183	1,0239	-0,0045	0,1171	1,0190	-0,0033	0,1185	1,0094
-0,0037	0,1197	1,0476	-0,0046	0,1164	1,0089	-0,0039	0,1198	1,0317
-0,0032	0,1179	1,0237	-0,0054	0,1188	1,0379	-0,0034	0,1182	1,0106
-0,0027	0,1161	0,9985	-0,0044	0,1163	1,0034	-0,0036	0,1196	1,0331
-0,0031	0,1180	1,0235	-0,0048	0,1175	1,0222	-0,0032	0,1182	1,0123
-0,0037	0,1199	1,0473	-0,0047	0,1179	1,0282	-0,0032	0,1196	1,0341
-0,0032	0,1179	1,0230	-0,0043	0,1160	0,9989	-0,0029	0,1180	1,0153
-0,0030	0,1161	0,9977	-0,0051	0,1185	1,0322	-0,0038	0,1193	1,0336
-0,0039	0,1180	1,0230	-0,0047	0,1171	1,0146	-0,0035	0,1179	1,0159
-0,0042	0,1202	1,0476	-0,0046	0,1167	1,0095	-0,0036	0,1194	1,0363
-0,0039	0,1185	1,0239	-0,0056	0,1188	1,0376	-0,0032	0,1179	1,0174
-0,0035	0,1167	0,9987	-0,0047	0,1166	1,0036	-0,0037	0,1189	1,0340
-0,0040	0,1187	1,0236	-0,0051	0,1178	1,0223	-0,0039	0,1175	1,0154
-0,0041	0,1210	1,0476	-0,0055	0,1182	1,0284	-0,0046	0,1185	1,0321
-0,0037	0,1188	1,0227	-0,0048	0,1163	0,9985	-0,0040	0,1167	1,0118
-0,0035	0,1171	0,9970	-0,0054	0,1188	1,0319	-0,0044	0,1178	1,0308
-0,0043	0,1189	1,0224	-0,0052	0,1174	1,0130	-0,0041	0,1168	1,0145
-0,0047	0,1207	1,0469	-0,0047	0,1167	1,0065	-0,0043	0,1181	1,0336
-0,0044	0,1184	1,0215	-0,0054	0,1187	1,0352	-0,0038	0,1168	1,0144
-0,0043	0,1166	0,9961	-0,0044	0,1163	1,0004	-0,0044	0,1178	1,0327
-0,0044	0,1185	1,0219	-0,0044	0,1176	1,0211	-0,0040	0,1165	1,0137
-0,0044	0,1202	1,0463	-0,0044	0,1183	1,0297	-0,0045	0,1175	1,0308
-0,0041	0,1185	1,0220	-0,0037	0,1163	1,0008	-0,0040	0,1161	1,0121
-0,0037	0,1167	0,9967	-0,0048	0,1187	1,0341	-0,0045	0,1172	1,0254
-0,0047	0,1188	1,0216	-0,0045	0,1175	1,0150	-0,0042	0,1163	1,0113
-0,0052	0,1207	1,0461	-0,0043	0,1173	1,0093	-0,0046	0,1169	1,0244
-0,0044	0,1186	1,0210	-0,0049	0,1196	1,0374	-0,0041	0,1160	1,0106
-0,0042	0,1167	0,9959	-0,0041	0,1172	1,0036	-0,0043	0,1168	1,0275
-0,0040	0,1187	1,0220	-0,0048	0,1185	1,0224	-0,0036	0,1157	1,0125
-0,0044	0,1205	1,0466	-0,0049	0,1191	1,0286	-0,0038	0,1166	1,0260
-0,0037	0,1184	1,0220	-0,0042	0,1171	0,9990	-0,0036	0,1157	1,0125
-0,0038	0,1162	0,9971	-0,0052	0,1198	1,0324	-0,0042	0,1167	1,0271
-0,0040	0,1185	1,0227	-0,0047	0,1186	1,0128	-0,0039	0,1157	1,0103
-0,0043	0,1203	1,0465	-0,0043	0,1179	1,0053	-0,0045	0,1169	1,0249
-0,0038	0,1186	1,0226	-0,0049	0,1201	1,0344	-0,0038	0,1159	1,0119
-0,0036	0,1166	0,9967	-0,0043	0,1172	0,9996	-0,0045	0,1170	1,0279
-0,0039	0,1185	1,0221	-0,0046	0,1183	1,0183	-0,0039	0,1158	1,0141
-0,0040	0,1204	1,0460	-0,0047	0,1183	1,0221	-0,0046	0,1167	1,0290
-0,0033	0,1186	1,0219	-0,0040	0,1165	0,9950	-0,0040	0,1157	1,0127
-0,0028	0,1170	0,9960	-0,0048	0,1192	1,0285	-0,0046	0,1165	1,0277
-0,0030	0,1189	1,0214	-0,0043	0,1178	1,0077	-0,0039	0,1159	1,0147
-0,0030	0,1207	1,0461	-0,0040	0,1176	1,0042	-0,0043	0,1168	1,0301

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 30 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

-0,0027	0,1186	1,0213	-0,0047	0,1196	1,0300	-0,0039	0,1162	1,0161
-0,0025	0,1167	0,9954	-0,0039	0,1169	0,9952	-0,0043	0,1175	1,0305
-0,0027	0,1188	1,0210	-0,0044	0,1189	1,0182	-0,0040	0,1165	1,0148
-0,0031	0,1205	1,0454	-0,0042	0,1190	1,0230	-0,0045	0,1176	1,0288
-0,0027	0,1183	1,0213	-0,0037	0,1168	0,9964	-0,0040	0,1164	1,0121
-0,0026	0,1163	0,9953	-0,0045	0,1195	1,0299	-0,0046	0,1173	1,0262
-0,0032	0,1181	1,0211	-0,0041	0,1177	1,0079	-0,0040	0,1167	1,0131
-0,0036	0,1198	1,0452	-0,0037	0,1174	1,0036	-0,0045	0,1175	1,0284
-0,0029	0,1180	1,0212	-0,0044	0,1194	1,0302	-0,0043	0,1165	1,0124
-0,0028	0,1161	0,9959	-0,0035	0,1171	0,9956	-0,0051	0,1176	1,0265
-0,0031	0,1179	1,0214	-0,0042	0,1193	1,0189	-0,0047	0,1168	1,0100
-0,0032	0,1197	1,0455	-0,0048	0,1192	1,0240	-0,0049	0,1178	1,0245
-0,0026	0,1179	1,0205	-0,0041	0,1173	1,0001	-0,0047	0,1170	1,0119
-0,0026	0,1159	0,9944	-0,0050	0,1199	1,0335	-0,0054	0,1180	1,0256
-0,0033	0,1176	1,0201	-0,0043	0,1182	1,0090	-0,0049	0,1167	1,0111
-0,0037	0,1191	1,0435	-0,0045	0,1183	1,0081	-0,0053	0,1178	1,0248
-0,0031	0,1171	1,0182	-0,0051	0,1200	1,0321	-0,0049	0,1167	1,0099
-0,0026	0,1155	0,9927	-0,0043	0,1173	0,9975	-0,0054	0,1177	1,0242
-0,0030	0,1175	1,0194	-0,0049	0,1190	1,0231	-0,0052	0,1168	1,0080
-0,0035	0,1195	1,0443	-0,0050	0,1190	1,0235	-0,0055	0,1178	1,0217
-0,0030	0,1173	1,0194	-0,0043	0,1175	0,9996	-0,0052	0,1166	1,0072
-0,0025	0,1154	0,9932	-0,0052	0,1201	1,0330	-0,0058	0,1174	1,0201
-0,0027	0,1175	1,0192	-0,0048	0,1183	1,0096	-0,0052	0,1163	1,0044
-0,0029	0,1192	1,0442	-0,0049	0,1183	1,0093	-0,0056	0,1173	1,0186
-0,0026	0,1171	1,0193	-0,0055	0,1199	1,0333	-0,0054	0,1165	1,0026
-0,0024	0,1154	0,9938	-0,0046	0,1172	0,9990	-0,0054	0,1175	1,0164
-0,0031	0,1174	1,0203	-0,0051	0,1190	1,0255	-0,0052	0,1163	1,0022
-0,0035	0,1194	1,0449	-0,0051	0,1187	1,0208	-0,0054	0,1172	1,0150
-0,0031	0,1177	1,0205	-0,0047	0,1173	1,0011	-0,0049	0,1160	1,0001
-0,0031	0,1156	0,9950	-0,0055	0,1197	1,0344	-0,0054	0,1171	1,0144
-0,0032	0,1175	1,0209	-0,0044	0,1173	1,0042	-0,0050	0,1162	0,9986
-0,0035	0,1195	1,0458	-0,0046	0,1185	1,0147	-0,0055	0,1171	1,0123
-0,0032	0,1178	1,0210	-0,0049	0,1195	1,0311	-0,0050	0,1163	0,9992
-0,0028	0,1159	0,9962	-0,0044	0,1173	0,9974	-0,0052	0,1171	1,0126
-0,0029	0,1178	1,0218	-0,0050	0,1198	1,0287	-0,0050	0,1160	1,0017
-0,0031	0,1197	1,0464	-0,0047	0,1192	1,0208	-0,0055	0,1165	1,0106
-0,0027	0,1176	1,0213	-0,0043	0,1180	1,0043	-0,0055	0,1163	1,0019
-0,0025	0,1157	0,9953	-0,0049	0,1204	1,0367	-0,0056	0,1174	1,0140
-0,0026	0,1175	1,0214	-0,0042	0,1182	1,0061	-0,0058	0,1173	1,0052
-0,0030	0,1194	1,0460	-0,0038	0,1191	1,0160	-0,0057	0,1180	1,0172
-0,0024	0,1174	1,0210	-0,0044	0,1201	1,0312	-0,0063	0,1173	1,0075
-0,0020	0,1157	0,9949	-0,0039	0,1176	0,9975	-0,0067	0,1180	1,0200
-0,0023	0,1179	1,0212	-0,0048	0,1201	1,0287	-0,0065	0,1169	1,0094
-0,0026	0,1201	1,0460	-0,0042	0,1193	1,0177	-0,0067	0,1176	1,0225

-0,0019	0,1181	1,0210	-0,0039	0,1182	1,0022	-0,0062	0,1165	1,0116
-0,0017	0,1163	0,9965	-0,0044	0,1205	1,0354	-0,0061	0,1170	1,0207
-0,0020	0,1185	1,0224	-0,0036	0,1183	1,0042	-0,0057	0,1160	1,0106
-0,0022	0,1204	1,0464	-0,0040	0,1188	1,0129	-0,0057	0,1169	1,0226
-0,0019	0,1187	1,0203	-0,0043	0,1198	1,0295	-0,0059	0,1165	1,0132
-0,0018	0,1166	0,9964	-0,0031	0,1176	0,9971	-0,0058	0,1176	1,0240
-0,0022	0,1182	1,0232	-0,0038	0,1199	1,0298	-0,0051	0,1165	1,0111
-0,0023	0,1199	1,0464	-0,0035	0,1189	1,0173	-0,0050	0,1175	1,0229
-0,0018	0,1178	1,0201	-0,0032	0,1178	1,0050	-0,0053	0,1173	1,0132
-0,0018	0,1162	0,9965	-0,0040	0,1200	1,0376	-0,0055	0,1180	1,0230
-0,0022	0,1180	1,0235	-0,0034	0,1173	1,0052	-0,0053	0,1169	1,0102
-0,0024	0,1199	1,0464	-0,0038	0,1187	1,0169	-0,0055	0,1176	1,0199
-0,0019	0,1181	1,0198	-0,0041	0,1197	1,0299	-0,0051	0,1166	1,0088
-0,0018	0,1165	0,9975	-0,0035	0,1171	0,9977	-0,0052	0,1171	1,0189
-0,0022	0,1191	1,0261	-0,0044	0,1195	1,0309	-0,0047	0,1161	1,0070
-0,0027	0,1206	1,0471	-0,0038	0,1185	1,0182	-0,0048	0,1169	1,0176
-0,0021	0,1187	1,0197	-0,0035	0,1177	1,0048	-0,0045	0,1162	1,0067
-0,0020	0,1170	0,9972	-0,0042	0,1204	1,0373	-0,0047	0,1170	1,0199
-0,0031	0,1193	1,0263	-0,0037	0,1181	1,0044	-0,0042	0,1160	1,0097
-0,0036	0,1206	1,0479	-0,0042	0,1194	1,0167	-0,0045	0,1168	1,0224
-0,0034	0,1187	1,0212	-0,0046	0,1202	1,0296	-0,0040	0,1160	1,0114
-0,0032	0,1170	0,9987	-0,0037	0,1178	0,9973	-0,0042	0,1164	1,0199
-0,0040	0,1192	1,0278	-0,0045	0,1202	1,0300	-0,0042	0,1158	1,0103
-0,0044	0,1203	1,0480	-0,0041	0,1194	1,0173	-0,0043	0,1167	1,0223
-0,0038	0,1178	1,0193	-0,0037	0,1183	1,0051	-0,0047	0,1164	1,0134
-0,0036	0,1163	0,9987	-0,0041	0,1205	1,0368	-0,0048	0,1171	1,0251
-0,0045	0,1185	1,0280	-0,0033	0,1182	1,0039	-0,0046	0,1163	1,0148
-0,0045	0,1201	1,0472	-0,0035	0,1192	1,0205	-0,0050	0,1171	1,0268
-0,0039	0,1179	1,0187	-0,0031	0,1197	1,0303	-0,0050	0,1172	1,0177
-0,0036	0,1164	0,9984	-0,0023	0,1175	1,0001	-0,0051	0,1181	1,0276
-0,0040	0,1187	1,0286	-0,0033	0,1199	1,0332	-0,0048	0,1169	1,0156
-0,0041	0,1199	1,0481	-0,0031	0,1187	1,0167	-0,0049	0,1179	1,0261
-0,0038	0,1176	1,0191	-0,0032	0,1178	1,0050	-0,0055	0,1176	1,0165
-0,0038	0,1159	0,9984	-0,0040	0,1200	1,0355	-0,0056	0,1183	1,0285
-0,0044	0,1186	1,0278	-0,0031	0,1175	1,0013	-0,0051	0,1176	1,0167
-0,0047	0,1203	1,0476	-0,0036	0,1187	1,0190	-0,0054	0,1183	1,0264
-0,0040	0,1183	1,0181	-0,0038	0,1194	1,0261	-0,0051	0,1173	1,0146
-0,0038	0,1170	0,9979	-0,0033	0,1172	0,9959	-0,0051	0,1182	1,0246
-0,0046	0,1191	1,0270	-0,0040	0,1197	1,0294	-0,0048	0,1173	1,0127
-0,0047	0,1205	1,0463	-0,0040	0,1185	1,0117	-0,0048	0,1177	1,0232
-0,0039	0,1180	1,0177	-0,0036	0,1180	1,0057	-0,0045	0,1166	1,0128
-0,0037	0,1167	0,9975	-0,0043	0,1202	1,0340	-0,0050	0,1174	1,0246
-0,0042	0,1190	1,0281	-0,0033	0,1176	0,9999	-0,0052	0,1165	1,0150
-0,0042	0,1201	1,0477	-0,0039	0,1188	1,0197	-0,0057	0,1177	1,0275

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 32 dari 58

-0,0037	0,1177	1,0191	-0,0036	0,1193	1,0290	-0,0055	0,1168	1,0166
-0,0033	0,1162	0,9989	-0,0030	0,1173	1,0004	-0,0057	0,1175	1,0254
-0,0043	0,1185	1,0281	-0,0040	0,1196	1,0337	-0,0055	0,1165	1,0149
-0,0046	0,1200	1,0484	-0,0037	0,1184	1,0159	-0,0056	0,1171	1,0261
-0,0039	0,1175	1,0184	-0,0034	0,1178	1,0105	-0,0057	0,1168	1,0163
-0,0038	0,1161	0,9980	-0,0042	0,1199	1,0393	-0,0059	0,1176	1,0289
-0,0046	0,1188	1,0274	-0,0035	0,1173	1,0047	-0,0055	0,1166	1,0178
-0,0047	0,1202	1,0471	-0,0043	0,1186	1,0228	-0,0059	0,1173	1,0298
-0,0041	0,1179	1,0183	-0,0041	0,1192	1,0315	-0,0057	0,1168	1,0254
-0,0037	0,1165	0,9980	-0,0037	0,1172	1,0024	-0,0059	0,1171	1,0309
-0,0045	0,1190	1,0268	-0,0046	0,1195	1,0357	-0,0052	0,1173	1,0224
-0,0048	0,1206	1,0458	-0,0039	0,1179	1,0156	-0,0054	0,1177	1,0284
-0,0041	0,1186	1,0168	-0,0035	0,1176	1,0081	-0,0049	0,1169	1,0238
-0,0037	0,1169	0,9958	-0,0044	0,1197	1,0365	-0,0052	0,1172	1,0311
-0,0046	0,1192	1,0254	-0,0037	0,1173	1,0022	-0,0054	0,1169	1,0218
-0,0046	0,1205	1,0448	-0,0043	0,1184	1,0204	-0,0054	0,1172	1,0264
-0,0042	0,1183	1,0159	-0,0043	0,1189	1,0265	-0,0054	0,1176	1,0241

2.1.2.2.6 Tes Sumbu Z pada 1,5 G

10 Hz			15 Hz			20 Hz		
Acc X	Acc Y	Acc Z	Acc X2	Acc Y3	Acc Z4	Acc X3	Acc Y4	Acc Z5
-0,0029	0,1149	0,8418	-0,0021	0,1125	0,9289	-0,0049	0,1151	1,0164
-0,0037	0,1190	0,8804	-0,0026	0,1126	0,9475	-0,0041	0,1165	1,0308
-0,0034	0,1208	0,9131	-0,0016	0,1095	0,9066	-0,0039	0,1154	1,0153
-0,0026	0,1169	0,8652	-0,0030	0,1136	0,9430	-0,0050	0,1159	1,0309
-0,0026	0,1145	0,8404	-0,0027	0,1124	0,9260	-0,0046	0,1148	1,0129
-0,0032	0,1189	0,8791	-0,0023	0,1115	0,9120	-0,0045	0,1170	1,0337
-0,0035	0,1207	0,9092	-0,0057	0,1094	0,9483	-0,0042	0,1156	1,0138
-0,0023	0,1165	0,8601	-0,0045	0,1063	0,9037	-0,0045	0,1178	1,0320
-0,0025	0,1145	0,8430	-0,0050	0,1093	0,9335	-0,0042	0,1164	1,0126
0,0484	0,1095	0,8815	-0,0055	0,1101	0,9503	-0,0056	0,1179	1,0284
0,0461	0,1113	0,9089	-0,0045	0,1074	0,9109	-0,0051	0,1165	1,0099
0,0452	0,1076	0,8588	-0,0057	0,1113	0,9472	-0,0049	0,1179	1,0319
0,0439	0,1057	0,8459	-0,0049	0,1097	0,9239	-0,0046	0,1162	1,0088
-0,0217	0,1044	0,8375	-0,0044	0,1090	0,9127	-0,0044	0,1174	1,0314
-0,0204	0,1050	0,8613	-0,0051	0,1118	0,9490	-0,0042	0,1156	1,0086
-0,0191	0,1015	0,8123	-0,0039	0,1085	0,9034	-0,0036	0,1167	1,0309
-0,0188	0,1005	0,7988	-0,0042	0,1110	0,9344	-0,0033	0,1149	1,0085
-0,0230	0,1064	0,8011	-0,0041	0,1116	0,9516	-0,0040	0,1165	1,0324
-0,0212	0,1070	0,8258	-0,0036	0,1087	0,9118	-0,0033	0,1146	1,0098
-0,0199	0,1034	0,7780	-0,0046	0,1128	0,9481	-0,0032	0,1161	1,0322

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 33 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

-0,0192	0,1017	0,7687	-0,0038	0,1109	0,9236	-0,0028	0,1139	1,0092
-0,0275	0,1200	0,8097	-0,0035	0,1102	0,9142	-0,0025	0,1153	1,0315
-0,0269	0,1206	0,8272	-0,0030	0,1134	0,9504	-0,0021	0,1137	1,0097
-0,0252	0,1166	0,7792	-0,0023	0,1101	0,9028	-0,0029	0,1155	1,0329
-0,0250	0,1169	0,7745	-0,0032	0,1126	0,9372	-0,0026	0,1134	1,0102
-0,0267	0,1164	0,8153	-0,0033	0,1127	0,9438	-0,0027	0,1150	1,0324
-0,0264	0,1173	0,8334	-0,0022	0,1101	0,9110	-0,0022	0,1131	1,0091
-0,0247	0,1135	0,7853	-0,0688	0,1215	0,8723	-0,0021	0,1148	1,0313
-0,0243	0,1136	0,7810	-0,0657	0,1192	0,8405	-0,0018	0,1131	1,0089
-0,0268	0,1214	0,8216	-0,0637	0,1194	0,8494	-0,0027	0,1149	1,0320
-0,0254	0,1225	0,8417	-0,0634	0,1197	0,8869	-0,0025	0,1128	1,0095
-0,0239	0,1186	0,7927	-0,0601	0,1160	0,8414	-0,0022	0,1148	1,0317
-0,0239	0,1180	0,7869	-0,0588	0,1193	0,8800	-0,0019	0,1126	1,0090
-0,0276	0,1108	0,8273	-0,0574	0,1185	0,8845	-0,0017	0,1143	1,0312
-0,0270	0,1121	0,8471	-0,0552	0,1162	0,8562	-0,0014	0,1123	1,0089
-0,0253	0,1085	0,7986	-0,0523	0,0729	0,8943	-0,0021	0,1141	1,0321
-0,0254	0,1090	0,7928	-0,0500	0,0714	0,8609	-0,0018	0,1125	1,0075
-0,0258	0,1086	0,8331	-0,0481	0,0731	0,8652	-0,0023	0,1147	1,0303
-0,0258	0,1098	0,8523	-0,0468	0,0763	0,9031	-0,0021	0,1132	1,0094
-0,0240	0,1060	0,8034	-0,0443	0,0741	0,8576	-0,0024	0,1149	1,0318
-0,0240	0,1058	0,7972	-0,0436	0,0785	0,8956	-0,0023	0,1135	1,0077
-0,0287	0,1204	0,8373	-0,0418	0,0793	0,8943	-0,0023	0,1152	1,0300
-0,0288	0,1213	0,8562	-0,0400	0,0783	0,8698	-0,0023	0,1132	1,0074
-0,0271	0,1171	0,8070	-0,0158	0,0714	0,9075	-0,0027	0,1154	1,0303
-0,0269	0,1172	0,8011	-0,0144	0,0701	0,8721	-0,0021	0,1139	1,0093
-0,0275	0,1157	0,8411	-0,0137	0,0738	0,8844	-0,0024	0,1157	1,0321
-0,0275	0,1168	0,8601	-0,0130	0,0771	0,9216	-0,0022	0,1140	1,0082
-0,0258	0,1129	0,8105	-0,0115	0,0748	0,8754	-0,0021	0,1155	1,0304
-0,0253	0,1127	0,8059	-0,0119	0,0793	0,9129	-0,0013	0,1137	1,0078
-0,0265	0,1138	0,8457	-0,0114	0,0805	0,9101	-0,0009	0,1153	1,0298
-0,0268	0,1151	0,8633	-0,0103	0,0798	0,8833	-0,0008	0,1132	1,0080
-0,0252	0,1112	0,8141	0,0130	0,0725	0,9205	-0,0007	0,1148	1,0297
-0,0250	0,1111	0,8074	0,0131	0,0708	0,8854	-0,0004	0,1128	1,0078
-0,0330	0,1292	0,8472	0,0122	0,0737	0,8964	-0,0010	0,1148	1,0307
-0,0330	0,1302	0,8655	0,0107	0,0776	0,9276	-0,0006	0,1127	1,0084
-0,0314	0,1257	0,8163	0,0112	0,0753	0,8810	-0,0006	0,1141	1,0309
-0,0310	0,1254	0,8105	0,0098	0,0796	0,9183	-0,0003	0,1125	1,0082
-0,0274	0,1289	0,8502	0,0096	0,0807	0,9161	-0,0011	0,1143	1,0320
-0,0277	0,1295	0,8679	0,0100	0,0795	0,8900	-0,0008	0,1124	1,0088
-0,0261	0,1252	0,8184	0,0196	0,0815	0,9270	-0,0007	0,1144	1,0308
-0,0256	0,1244	0,8143	0,0197	0,0796	0,8909	-0,0003	0,1127	1,0099
-0,0288	0,1182	0,8538	0,0186	0,0811	0,9006	-0,0008	0,1149	1,0328
-0,0288	0,1190	0,8723	0,0184	0,0841	0,9372	-0,0004	0,1135	1,0089
-0,0267	0,1149	0,8229	0,0187	0,0817	0,8910	-0,0004	0,1158	1,0317

-0,0266	0,1148	0,8181	0,0171	0,0860	0,9280	-0,0004	0,1144	1,0078
-0,0309	0,1066	0,8575	0,0164	0,0869	0,9259	-0,0008	0,1167	1,0308
-0,0289	0,1084	0,8780	0,0163	0,0856	0,9004	-0,0006	0,1147	1,0091
-0,0273	0,1047	0,8283	0,0375	0,0773	0,9370	-0,0009	0,1167	1,0312
-0,0269	0,1050	0,8228	0,0367	0,0758	0,9010	-0,0007	0,1152	1,0074
-0,0299	0,0996	0,8621	0,0353	0,0792	0,9125	-0,0015	0,1173	1,0300
-0,0295	0,1011	0,8810	0,0341	0,0819	0,9488	-0,0010	0,1160	1,0054
-0,0278	0,0973	0,8313	0,0338	0,0797	0,9023	-0,0018	0,1181	1,0281
-0,0275	0,0980	0,8246	0,0316	0,0838	0,9389	-0,0016	0,1160	1,0060
-0,0326	0,1136	0,8638	0,0305	0,0844	0,9354	-0,0022	0,1176	1,0292
-0,0321	0,1149	0,8821	0,0301	0,0829	0,9074	-0,0017	0,1158	1,0071
-0,0304	0,1109	0,8324	0,0429	0,0862	0,9438	-0,0022	0,1179	1,0299
-0,0301	0,1110	0,8273	0,0422	0,0844	0,9070	-0,0021	0,1165	1,0089
-0,0354	0,0995	0,8664	0,0407	0,0876	0,9184	-0,0029	0,1183	1,0317
-0,0337	0,1014	0,8855	0,0382	0,0913	0,9484	-0,0025	0,1165	1,0094
-0,0317	0,0981	0,8352	0,0378	0,0886	0,9021	-0,0030	0,1182	1,0332
-0,0313	0,0979	0,8304	0,0351	0,0928	0,9387	-0,0028	0,1168	1,0075
-0,0274	0,1118	0,8694	0,0344	0,0937	0,9324	-0,0035	0,1187	1,0344
-0,0267	0,1124	0,8818	0,0335	0,0925	0,9083	-0,0032	0,1166	1,0090
-0,0252	0,1083	0,8319	0,0240	0,0828	0,9447	-0,0039	0,1180	1,0341
-0,0236	0,1070	0,8292	0,0242	0,0803	0,9038	-0,0037	0,1159	1,0096
-0,0219	0,1176	0,8682	0,0232	0,0819	0,9239	-0,0044	0,1176	1,0354
-0,0216	0,1179	0,8832	0,0221	0,0874	0,9532	-0,0038	0,1161	1,0087
-0,0201	0,1137	0,8335	0,0224	0,0847	0,9048	-0,0044	0,1177	1,0334
-0,0195	0,1128	0,8299	0,0204	0,0893	0,9413	-0,0039	0,1162	1,0073
-0,0162	0,1261	0,8689	0,0201	0,0904	0,9352	-0,0046	0,1180	1,0330
-0,0156	0,1262	0,8816	0,0198	0,0894	0,9110	-0,0045	0,1163	1,0083
-0,0145	0,1218	0,8316	0,0159	0,0923	0,9440	-0,0054	0,1182	1,0338
-0,0142	0,1210	0,8318	0,0164	0,0899	0,9040	-0,0047	0,1167	1,0078
-0,0114	0,1331	0,8708	0,0159	0,0901	0,9223	-0,0054	0,1180	1,0325
-0,0113	0,1332	0,8858	0,0140	0,0940	0,9526	-0,0046	0,1162	1,0069
-0,0100	0,1284	0,8361	0,0146	0,0912	0,9045	-0,0054	0,1178	1,0326
-0,0086	0,1265	0,8341	0,0129	0,0956	0,9410	-0,0050	0,1162	1,0070
-0,0073	0,1355	0,8729	0,0128	0,0955	0,9314	-0,0057	0,1180	1,0330
-0,0076	0,1353	0,8876	0,0129	0,0941	0,9063	-0,0052	0,1164	1,0083
-0,0068	0,1304	0,8373	0,0047	0,0862	0,9428	-0,0057	0,1183	1,0340
-0,0054	0,1281	0,8339	0,0053	0,0839	0,9022	-0,0053	0,1164	1,0074
-0,0040	0,1381	0,8728	0,0046	0,0846	0,9180	-0,0060	0,1185	1,0347
-0,0048	0,1373	0,8863	0,0029	0,0888	0,9482	-0,0054	0,1164	1,0069
-0,0038	0,1326	0,8358	0,0039	0,0859	0,9015	-0,0060	0,1187	1,0363
-0,0040	0,1305	0,8316	0,0024	0,0903	0,9381	-0,0054	0,1167	1,0102
-0,0028	0,1416	0,8706	0,0024	0,0911	0,9311	-0,0060	0,1189	1,0386
-0,0036	0,1407	0,8847	0,0025	0,0898	0,9072	-0,0051	0,1169	1,0090
-0,0028	0,1356	0,8346	-0,0007	0,0933	0,9436	-0,0059	0,1192	1,0379

-0,0025	0,1335	0,8316	0,0003	0,0906	0,9029	-0,0052	0,1173	1,0102
-0,0001	0,1455	0,8705	0,0006	0,0910	0,9208	-0,0057	0,1197	1,0392
-0,0011	0,1447	0,8847	0,0001	0,0956	0,9509	-0,0050	0,1178	1,0115
-0,0002	0,1397	0,8346	0,0010	0,0928	0,9026	-0,0057	0,1200	1,0404
0,0009	0,1371	0,8320	-0,0003	0,0967	0,9391	-0,0048	0,1181	1,0133
0,0020	0,1462	0,8710	0,0001	0,0963	0,9285	-0,0052	0,1204	1,0425
0,0010	0,1451	0,8849	0,0009	0,0944	0,9037	-0,0046	0,1186	1,0147
0,0016	0,1399	0,8352	-0,0029	0,0961	0,9403	-0,0052	0,1208	1,0414
0,0009	0,1382	0,8300	-0,0021	0,0937	0,8989	-0,0044	0,1186	1,0126
0,0022	0,1481	0,8691	-0,0023	0,0942	0,9146	-0,0048	0,1208	1,0413
0,0013	0,1468	0,8832	-0,0023	0,0992	0,9445	-0,0042	0,1187	1,0139
0,0021	0,1415	0,8334	-0,0012	0,0963	0,8961	-0,0047	0,1210	1,0411
0,0016	0,1395	0,8290	-0,0025	0,1004	0,9329	-0,0040	0,1188	1,0120
0,0027	0,1505	0,8681	-0,0022	0,0996	0,9217	-0,0047	0,1209	1,0408
0,0021	0,1503	0,8810	-0,0018	0,0983	0,8980	-0,0039	0,1187	1,0116
0,0028	0,1451	0,8310	-0,0066	0,0926	0,9335	-0,0047	0,1209	1,0394
0,0031	0,1423	0,8276	-0,0051	0,0905	0,8913	-0,0040	0,1190	1,0099
0,0041	0,1521	0,8667	-0,0054	0,0921	0,9050	-0,0046	0,1211	1,0379
0,0030	0,1510	0,8813	-0,0058	0,0967	0,9359	-0,0040	0,1187	1,0086
0,0037	0,1457	0,8313	-0,0045	0,0939	0,8892	-0,0047	0,1210	1,0366
0,0032	0,1441	0,8326	-0,0058	0,0981	0,9262	-0,0041	0,1190	1,0077
0,0043	0,1549	0,8715	-0,0052	0,0983	0,9184	-0,0051	0,1211	1,0357
0,0030	0,1534	0,8852	-0,0049	0,0970	0,8949	-0,0045	0,1189	1,0070
0,0039	0,1482	0,8354	-0,0089	0,1087	0,9308	-0,0056	0,1209	1,0344
0,0042	0,1459	0,8322	-0,0079	0,1059	0,8894	-0,0052	0,1188	1,0057
0,0065	0,1575	0,8712	-0,0078	0,1051	0,9072	-0,0056	0,1212	1,0342
0,0064	0,1566	0,8836	-0,0085	0,1088	0,9375	-0,0047	0,1191	1,0048
0,0069	0,1510	0,8335	-0,0073	0,1054	0,8900	-0,0055	0,1212	1,0323
0,0076	0,1481	0,8309	-0,0082	0,1093	0,9270	-0,0047	0,1191	1,0037
0,0082	0,1567	0,8699	-0,0075	0,1091	0,9193	-0,0055	0,1214	1,0327
0,0074	0,1559	0,8846	-0,0073	0,1076	0,8975	-0,0046	0,1192	1,0053
0,0078	0,1506	0,8348	-0,0126	0,1054	0,9342	-0,0052	0,1215	1,0342
0,0072	0,1485	0,8357	-0,0113	0,1024	0,8943	-0,0046	0,1191	1,0048
0,0085	0,1580	0,8745	-0,0110	0,1019	0,9115	-0,0050	0,1212	1,0359
0,0072	0,1565	0,8884	-0,0116	0,1059	0,9420	-0,0044	0,1187	1,0055
0,0079	0,1510	0,8380	-0,0103	0,1024	0,8945	-0,0051	0,1210	1,0353
0,0073	0,1487	0,8334	-0,0110	0,1060	0,9314	-0,0045	0,1183	1,0044
0,0083	0,1597	0,8723	-0,0101	0,1059	0,9241	-0,0052	0,1206	1,0341
0,0082	0,1589	0,8850	-0,0096	0,1042	0,9013	-0,0045	0,1180	1,0035
0,0085	0,1534	0,8354	-0,0172	0,0940	0,9379	-0,0050	0,1205	1,0351
0,0086	0,1512	0,8361	-0,0160	0,0916	0,8951	-0,0042	0,1176	1,0025
0,0058	0,1525	0,8749	-0,0158	0,0929	0,9098	-0,0048	0,1201	1,0358
0,0053	0,1509	0,8828	-0,0161	0,0969	0,9405	-0,0041	0,1176	1,0044
0,0056	0,1458	0,8335	-0,0146	0,0942	0,8931	-0,0044	0,1200	1,0370

Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 36 dari 58

© 2017 Prodi Teknik Elektro-ITB. Pengungkapan dan penggunaan seluruh isi dokumen hanya dapat dilakukan atas izin tertulis Prodi Teknik Elektro - ITB Jalan Ganesha 10 Bandung, 40132 Indonesia.

0,0053	0,1453	0,8400	-0,0154	0,0983	0,9300	-0,0038	0,1170	1,0041
0,0027	0,1495	0,8787	-0,0144	0,0987	0,9237	-0,0043	0,1195	1,0366
0,0028	0,1481	0,8837	-0,0136	0,0972	0,9001	-0,0036	0,1169	1,0045
0,0032	0,1426	0,8340	-0,0123	0,1051	0,9368	-0,0042	0,1195	1,0376
0,0026	0,1417	0,8394	-0,0113	0,1023	0,8942	-0,0034	0,1168	1,0055
0,0018	0,1455	0,8781	-0,0111	0,1046	0,9080	-0,0039	0,1195	1,0382
0,0019	0,1446	0,8848	-0,0115	0,1086	0,9383	-0,0032	0,1168	1,0054
0,0026	0,1393	0,8357	-0,0102	0,1050	0,8914	-0,0038	0,1194	1,0386
0,0018	0,1386	0,8408	-0,0111	0,1086	0,9283	-0,0033	0,1170	1,0048
0,0020	0,1422	0,8795	-0,0105	0,1077	0,9140	-0,0039	0,1196	1,0380
0,0023	0,1424	0,8833	-0,0099	0,1063	0,8937	-0,0032	0,1172	1,0042
0,0031	0,1378	0,8353	-0,0080	0,1181	0,9305	-0,0039	0,1197	1,0374
0,0006	0,1380	0,8451	-0,0063	0,1142	0,8872	-0,0035	0,1173	1,0031
0,0013	0,1419	0,8836	-0,0070	0,1166	0,9104	-0,0042	0,1198	1,0363
0,0010	0,1421	0,8857	-0,0071	0,1195	0,9383	-0,0036	0,1176	1,0021
0,0017	0,1375	0,8368	-0,0060	0,1157	0,8936	-0,0042	0,1203	1,0353
-0,0004	0,1373	0,8462	-0,0073	0,1194	0,9305	-0,0031	0,1180	1,0009
-0,0003	0,1408	0,8847	-0,0068	0,1180	0,9183	-0,0037	0,1207	1,0342
0,0000	0,1410	0,8877	-0,0063	0,1166	0,8980	-0,0033	0,1183	0,9999
0,0008	0,1364	0,8390	-0,0043	0,1283	0,9348	-0,0040	0,1205	1,0333
0,0021	0,1357	0,8490	-0,0037	0,1245	0,8897	-0,0033	0,1183	0,9989
0,0023	0,1397	0,8874	-0,0046	0,1261	0,9151	-0,0039	0,1211	1,0323
0,0025	0,1391	0,8921	-0,0052	0,1253	0,9348	-0,0033	0,1185	0,9979
0,0032	0,1348	0,8435	-0,0040	0,1218	0,8914	-0,0040	0,1209	1,0313
0,0010	0,1350	0,8531	-0,0051	0,1255	0,9283	-0,0033	0,1185	0,9974
0,0012	0,1387	0,8914	-0,0047	0,1237	0,9100	-0,0040	0,1212	1,0308
0,0008	0,1387	0,8929	-0,0045	0,1225	0,8976	-0,0030	0,1188	0,9965
0,0014	0,1341	0,8443	-0,0064	0,1255	0,9344	-0,0038	0,1214	1,0299
0,0022	0,1336	0,8538	-0,0052	0,1217	0,8884	-0,0028	0,1185	0,9959
0,0026	0,1375	0,8920	-0,0061	0,1234	0,9147	-0,0037	0,1211	1,0294
0,0026	0,1380	0,8941	-0,0070	0,1234	0,9370	-0,0028	0,1187	0,9953
0,0032	0,1330	0,8451	-0,0061	0,1200	0,8936	-0,0035	0,1212	1,0287
0,0034	0,1326	0,8550	-0,0074	0,1235	0,9305	-0,0025	0,1189	0,9939
0,0031	0,1367	0,8931	-0,0067	0,1221	0,9123	-0,0034	0,1214	1,0275
0,0027	0,1368	0,8944	-0,0065	0,1207	0,8995	-0,0025	0,1185	0,9924
0,0032	0,1322	0,8452	-0,0095	0,1193	0,9362	-0,0034	0,1211	1,0260
0,0019	0,1318	0,8546	-0,0080	0,1156	0,8911	-0,0026	0,1184	0,9912
0,0022	0,1357	0,8928	-0,0083	0,1178	0,9192	-0,0034	0,1211	1,0248
0,0021	0,1360	0,8951	-0,0090	0,1181	0,9419	-0,0026	0,1182	0,9893
0,0029	0,1314	0,8463	-0,0081	0,1148	0,8988	-0,0035	0,1206	1,0230
0,0004	0,1318	0,8556	-0,0090	0,1186	0,9355	-0,0025	0,1180	0,9880
0,0011	0,1354	0,8938	-0,0085	0,1170	0,9152	-0,0033	0,1208	1,0217
0,0017	0,1356	0,8965	-0,0081	0,1159	0,9017	-0,0024	0,1183	0,9866
0,0022	0,1311	0,8472	-0,0117	0,1127	0,9383	-0,0030	0,1209	1,0204

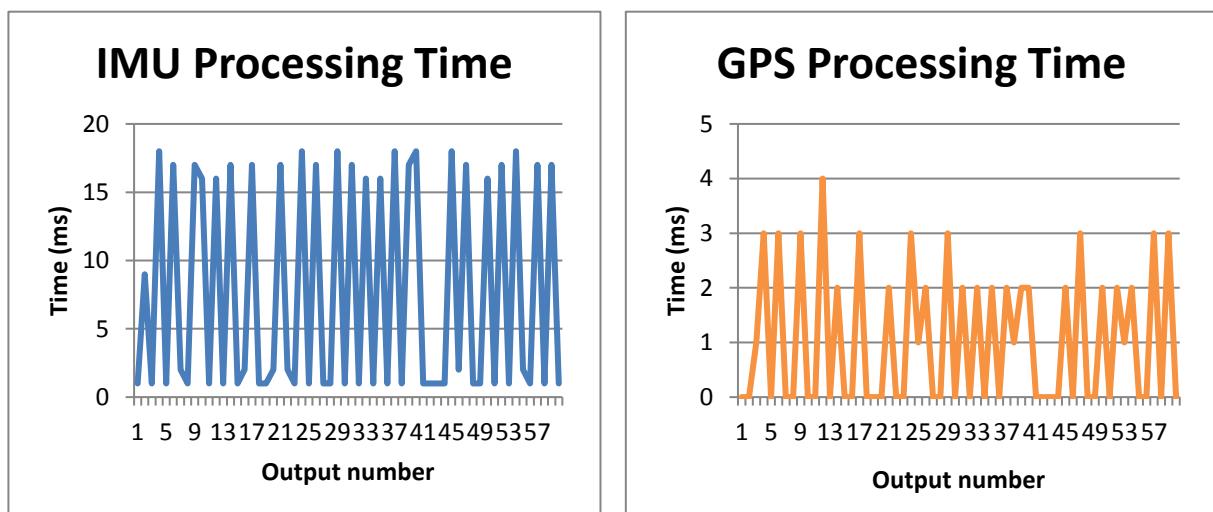
Nomor Dokumen: B500-02-TA1617.01. 069 Nomor Revisi:01 Tanggal: 27 April 2017 Halaman 37 dari 58

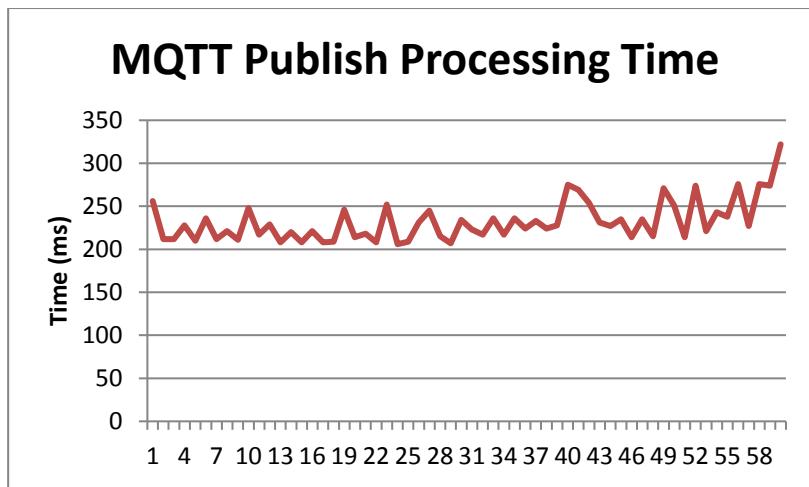
0,0033	0,1307	0,8567	-0,0102	0,1088	0,8917	-0,0022	0,1182	0,9858
0,0033	0,1351	0,8949	-0,0106	0,1111	0,9241	-0,0030	0,1209	1,0196
0,0032	0,1353	0,8972	-0,0103	0,1121	0,9415	-0,0021	0,1183	0,9842
0,0038	0,1308	0,8481	-0,0088	0,1093	0,8992	-0,0029	0,1212	1,0181

2.1.3 Pengujian Algoritma Sensor ECN

Pada pengujian algoritma sensor ECN, dilakukan dua jenis pengujian, yaitu pengujian waktu yang dibutuhkan dari beberapa proses penting pada algoritma yang didesain dan pengujian fungsional dari algoritma sensor ECN yang sudah didesain. Pada pengujian waktu proses, diamati tiga proses yaitu, proses pembacaan data IMU, pembacaan GPS, dan proses publish MQTT. Metodologi pengujian waktu proses ini dilakukan dengan memanfaatkan fungsi millis(). Sehingga pada algoritma sensor, Cuma ditambahkan fungsi millis sebelum dan setelah proses pembacaan data IMU, pembacaan GPS, dan publish MQTT.

Pengamatan ini dilakukan sebanyak 60 kali proses pengiriman data ke server setiap 1 detik dengan 40 data sampling IMU. Hasil pengujian waktu proses pada algoritma sensor dengan kondisi tanpa menunggu queue RabbitMQ dapat dilihat pada lampiran dan grafik dibawah ini.

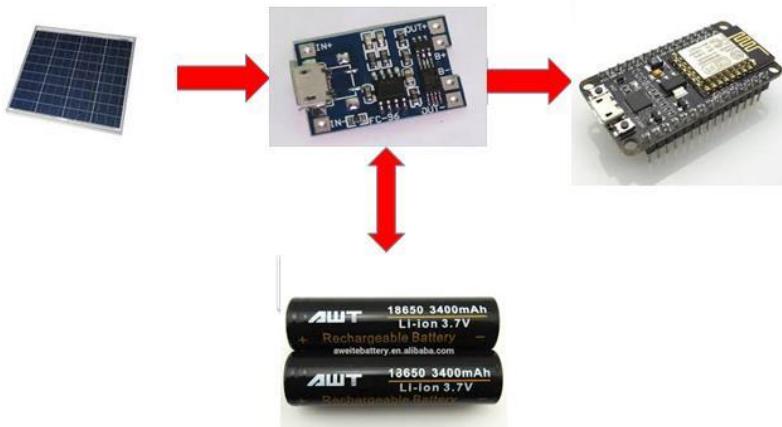




Dapat dilihat bahwa proses pembacaan GPS memakan waktu proses yang paling sebentar dengan range antara 0-4 ms, sehingga karena proses pembacaan GPS ini hanya dilakukan satu kali tiap satu detik maka lama proses pembacaan GPS tidak mengganggu sampling IMU. Sedangkan lama proses pembacaan IMU mengambil waktu antara 1-18 ms. Karena sensor harus dapat melakukan sampling dengan frekuensi 40 Hz atau setiap 25 ms maka lama proses pembacaan IMU yang dilakukan oleh sensor masih membuat sensor dapat melakukan sampling data IMU dengan periode 25 ms. Dapat dilihat bahwa dalam kondisi tidak menunggu queue RabbitMQ, proses pengiriman memakan waktu proses cukup lama yaitu diantara 206–322 ms. Hal tersebut cukup mengganggu proses pembacaan IMU, karena pada saat sensor publish message ke RabbitMQ, proses sampling data akan berhenti selama proses publish message tersebut. Hal tersebut dapat diatasi dengan mengimplementasikan RTOS(Real Time Operating System) pada board NodeMCU, tetapi proses implementasi terhambat karena library yang sangat terbatas jika menggunakan RTOS sehingga harus membuat library untuk pembacaan IMU, pembacaan GPS, dan MQTT yang memakan waktu lama.

2.1.4 Pengujian Daya Tahan Sistem Daya

Pengujian Daya Tahan Sistem Daya dilakukan dengan mencoba rangkaian charging solar panel yang terhubung dengan beban lalu diamati tegangan solar panel dalam beberapa waktu tertentu. Setelah itu, saat malam hari diuji apakah sistem daya yang didesain dapat menyuplai daya ke sensor dari malam hari sampai malam hari. Pada pengujian pertama digunakan 2 baterai UltraFire dengan kapasitas 6800 mAh per sel yang terkoneksi paralel dan solar panel dengan daya 5 watt. Rangkaian sistem daya dapat dilihat pada gambar berikut.



Pada pengujian ini, diamati tegangan pada 4 kondisi yang didefinisikan, yaitu cerah pagi untuk jam 07.00 pagi sampai jam 11.00 saat kondisi cerah, cerah siang untuk jam 11.00 sampai jam 16.00 saat kondisi cerah, gelap untuk jam 16.00 sampai jam 07.00 dan mendung ketika sinar matahari terhalang oleh awan saat siang hari. Hasil pengamatan menunjukkan bahwa tegangan rata-rata yang dihasilkan solar panel saat cerah pagi adalah 4,8 Volt, saat cerah siang tegangan rata-rata yang dihasilkan adalah 5,6 Volt, saat gelap tegangan rata-rata yang dihasilkan adalah 0,2 Volt, dan saat mendung tegangan rata-rata yang dihasilkan adalah 4,5 Volt. Berdasarkan datasheet modul charging baterai li-ion, input tegangan yang harus disuplai solar panel harus berada di range 4,5-5,5 Volt. Dapat disimpulkan dari data diatas bahwa modul dapat di-charge pada kondisi cerah pagi, cerah siang, dan mendung. Sehingga NodeMCU harus disuplai oleh baterai dari jam 16.00-07.00 pagi besok harinya.

Tetapi saat proses pengujian, tegangan baterai Li-ion menurun ke angka 2,8 Volt dari kondisi normal (3,7 Volt) karena terjadi short pada modul charging sehingga pengujian kapasitas baterai setelah di-charge pada siang hari tidak dapat dilakukan.

Pada pengujian berikutnya, komponen baterai diganti dengan 2 baterai AWT yang memiliki kapasitas 3400 mAh per sel yang terkoneksi secara paralel. Hasil pengujian dapat dilihat pada tabel dibawah ini.

Jam	Vsolar (V)	Vbatt (V)	Keterangan
12.00	5,27	3,495	Cerah berawan
12.31	3,92	3,559	Cerah terik
13.01	4,3	3,561	Cerah
13.32	3,92	3,551	Berawan

Dapat dilihat bahwa tegangan baterai terus meningkat seiring berjalananya waktu ketika di charge. Pengujian dihentikan pada pukul 13.52 karena terjadi hujan. Dari jam 13.52 tersebut sensor kami pasang pada sistem daya tersebut dan dapat bertahan sampai jam 03.02 dini hari. Hal ini mengindikasikan bahwa kualitas baterai AWT berbeda dengan baterai UltraFire karena tanpa disuplai oleh solar panel baterai AWT dapat bertahan selama 13 jam sedangkan baterai ultrafire tidak dapat menyuplai setelah solar panel berhenti menyuplai daya ke baterai. Tetapi setelah baterai tidak menyuplai sensor lagi, tegangan baterai menurun ke angka 2,8 Volt.

Dari proses pengujian, dapat disimpulkan bahwa dengan menggunakan baterai AWT, sistem daya dapat menyuplai daya sensor selama satu hari, tetapi setelah itu tegangan baterai menurun ke angka 2,8 Volt. Ketika tegangan baterai sudah menurun ke tingkat dibawah normal, maka diperlukan waktu yg lama untuk mengisi kembali kapasitas baterai tersebut. Sehingga dapat dikatakan bahwa desain sistem daya yang menyatakan bahwa dibutuhkan baterai dengan total kapasitas sekitar 12000 mAh dan daya solar panel sebesar 13,5 Watt atau dua kali kapasitas dan besar daya dari sistem daya pada saat pengujian.

2.2 Pengujian Messaging Server dan Database

Aspek yang ditinjau pada pengujian Messaging Server adalah pesan dapat diterima oleh Messaging Server dengan baik kemudian dapat meneruskan pesan tersebut ke consumer yang sesuai. Harus dilakukan sebuah pengaturan pada pengirim sehingga sebuah pesan tidak akan bisa dikonsumsi oleh consumer yang tidak sesuai. Setelah pesan diterima oleh consumer, pesan tersebut akan dikirimkan menuju sebuah database. Data yang telah disimpan dapat digunakan untuk diolah datanya.

Pengujian dilakukan dengan mengirimkan pesan dari sensor maupun dari program penguji. Pesan kemudian akan dikirimkan menuju Messaging Server. Supaya mengetahui bahwa pessan dapat terkirim, maka harus ada consumer yang menjalankan perintah consume Messaging Server. Apabila pesan dapat di consume, maka proses pengiriman dari sensor menuju Messaging Server berjalan dengan lancar.

2.2.1 Pengujian Messaging Server RabbitMQ

Sensor mengirimkan sebuah pesan setiap 1 detik menuju Messaging Server dengan jaringan WiFi yang sudah di set. Pesan ini merupakan pesan informasi geospasial dengan format JSON. Penguji melakukan debugging pada sensor dengan membaca komunikasi serial sehingga mengetahui apa yang dikirimkan oleh sensor.

Berikut ini data yang dihasilkan oleh sensor saat berhasil mengirimkan sebuah pesan beserta tampilan debugging dari pesan yang dihasilkan:



```
publish success
{
  "pointTime": "2000-0-0T0:0:0Z",
  "timeZone": "Asia/Jakarta",
  "interval": "1000",
  "clientID": "ECI"
}
publish success
{
  "pointTime": "2000-0-0T0:0:0Z",
  "timeZone": "Asia/Jakarta",
  "interval": "1000",
  "clientID": "ECI"
}
publish success
{
  "pointTime": "2000-0-0T0:0:0Z",
  "timeZone": "Asia/Jakarta",
  "interval": "1000",
  "clientID": "ECI"
}
publish success
{
  "pointTime": "2000-0-0T0:0:0Z",
  "timeZone": "Asia/Jakarta",
  "interval": "1000",
  "clientID": "ECI"
}
publish success
{
  "pointTime": "2000-0-0T0:0:0Z",
  "timeZone": "Asia/Jakarta",
  "interval": "1000",
  "clientID": "ECI"
}
publish success
{
  "pointTime": "2000-0-0T0:0:0Z",
  "timeZone": "Asia/Jakarta",
  "interval": "1000",
  "clientID": "ECI"
}
Soft WDT reset
```

Kemudian isi pesan yang dikirimkan adalah sebagai berikut:

```
{ "pointTime": "2000-0-0T0:0:0Z", "timeZone": "Asia/Jakarta", "interval": "1000", "clientID": "ECN-1", "geojson": {"geometry": {"type": "Point", "coordinates": [-6.89, 107.61]}, "properties": {"name": "ITB"}}, "accelerations": [{"x": -2.0661, "y": 4.8328, "z": 5.2528}, {"x": -2.0859, "y": 4.8904, "z": 5.3096}, {"x": -2.1068, "y": 4.9404, "z": 5.3676}, {"x": -2.1281, "y": 4.9897, "z": 5.4209}, {"x": -2.1516, "y": 5.0375, "z": 5.4726}, {"x": -2.1710, "y": 5.0862, "z": 5.5196}, {"x": -2.1899, "y": 5.1320, "z": 5.5701}, {"x": -2.2104, "y": 5.1750, "z": 5.6138}, {"x": -2.2282, "y": 5.2156, "z": 5.6570}, {"x": -2.2458, "y": 5.2601, "z": 5.7015}, {"x": -2.2648, "y": 5.3003, "z": 5.7426}, {"x": -2.2820, "y": 5.3374, "z": 5.7857}, {"x": -2.2963, "y": 5.3774, "z": 5.8231}, {"x": -2.3130, "y": 5.4122, "z": 5.8600}, {"x": -2.3297, "y": 5.4444, "z": 5.8985}, {"x": -2.3458, "y": 5.4768, "z": 5.9321}, {"x": -2.3593, "y": 5.5074, "z": 5.9641}, {"x": -2.3726, "y": 5.5376, "z": 5.9999}, {"x": -2.3879, "y": 5.5690, "z": 6.0290}, {"x": -2.3978, "y": 5.5987, "z": 6.0641}, {"x": -2.4110, "y": 5.6320, "z": 6.0913}, {"x": -2.4203, "y": 5.6601, "z": 6.1175}, {"x": -2.4339, "y": 5.6857, "z": 6.1457}, {"x": -2.4411, "y": 5.7115, "z": 6.1718}, {"x": -2.4504, "y": 5.7361, "z": 6.1995}, {"x": -2.4614, "y": 5.7576, "z": 6.2212}, {"x": -2.4707, "y": 5.7819, "z": 6.2485}, {"x": -2.4815, "y": 5.8033, "z": 6.2715}, {"x": -2.4923, "y": 5.8249, "z": 6.2971}, {"x": -2.5006, "y": 5.8451, "z": 6.3182}, {"x": -2.5088, "y": 5.8637, "z": 6.3422}, {"x": -2.5156, "y": 5.8826, "z": 6.3636}, {"x": -2.5221, "y": 5.9009, "z": 6.3845}, {"x": -2.5288, "y": 5.9186, "z": 6.4063}, {"x": -2.5368, "y": 5.9352, "z": 6.4273}, {"x": -2.5464, "y": 5.9518, "z": 6.4467}, {"x": -2.5534, "y": 5.9692, "z": 6.4634}, {"x": -2.5586, "y": 5.9802, "z": 6.4793}, {"x": -2.5656, "y": 5.9951, "z": 6.4893}, {"x": -2.5717, "y": 6.0085, "z": 6.5053}]} }
```

Pesan yang terkirim oleh sensor memiliki format yang sesuai dan data yang valid. Validitas pesan ini akan dicek kemudian oleh consumer.

2.2.2 Pengujian Database MySQL

Langkah selanjutnya, akan dicek apabila pesan tersebut berhasil diunduh dari Messaging Server. Ketika program consume oleh console C# dijalankan, maka hasilnya adalah sebagai berikut:

```

file:///C:/Users/X450JB/Desktop/Database/Database/bin/Debug/Database.EXE - □ X
Queue Declare Emergency GUI
Already BasicConsume
2017-04-26 14:27:31
Queue Declare Emergency GUI
Already BasicConsume
[x] Received { "pointTime": "2000-0-0T0:0:0Z", "timeZone": "Asia/Jakarta", "interval": "1000", "clientID": "ECN-1", "geojson": {"geometry": {"type": "Point", "coordinates": [-6.89, 107.61]}, "properties": {"name": "ITB"}}, "accelerations": [{"x": -2.6833, "y": 6.3658, "z": 7.0531}, {"x": -2.6821, "y": 6.3661, "z": 7.0531}, {"x": -2.6801, "y": 6.3652, "z": 7.0496}, {"x": -2.6790, "y": 6.3639, "z": 7.0497}, {"x": -2.6773, "y": 6.3652, "z": 7.0527}, {"x": -2.6770, "y": 6.3659, "z": 7.0548}, {"x": -2.6787, "y": 6.3653, "z": 7.0527}, {"x": -2.6760, "y": 6.3664, "z": 7.0510}, {"x": -2.6755, "y": 6.3678, "z": 7.0514}, {"x": -2.6736, "y": 6.3695, "z": 7.0526}, {"x": -2.6727, "y": 6.3698, "z": 7.0509}, {"x": -2.6731, "y": 6.3709, "z": 7.0499}, {"x": -2.6748, "y": 6.3696, "z": 7.0513}, {"x": -2.6740, "y": 6.3678, "z": 7.0528}, {"x": -2.6735, "y": 6.3664, "z": 7.0565}, {"x": -2.6715, "y": 6.3644, "z": 7.0574}, {"x": -2.6716, "y": 6.3644, "z": 7.0576}, {"x": -2.6697, "y": 6.3634, "z": 7.0555}, {"x": -2.6713, "y": 6.3650, "z": 7.0564}, {"x": -2.6724, "y": 6.3655, "z": 7.0586}, {"x": -2.6728, "y": 6.3644, "z": 7.0573}, {"x": -2.6733, "y": 6.3655, "z": 7.0554}, {"x": -2.6743, "y": 6.3655, "z": 7.0574}, {"x": -2.6726, "y": 6.3666, "z": 7.0578}, {"x": -2.6730, "y": 6.3660, "z": 7.0542}, {"x": -2.6716, "y": 6.3649, "z": 7.0543}, {"x": -2.6727, "y": 6.3645, "z": 7.0546}, {"x": -2.6738, "y": 6.3666, "z": 7.0531}, {"x": -2.6736, "y": 6.3661, "z": 7.0501}, {"x": -2.6738, "y": 6.3655, "z": 7.0512}, {"x": -2.6740, "y": 6.3645, "z": 7.0508}, {"x": -2.6754, "y": 6.3647, "z": 7.0511}, {"x": -2.6747, "y": 6.3663, "z": 7.0539}, {"x": -2.6742, "y": 6.3661, "z": 7.0555}, {"x": -2.6715, "y": 6.3672, "z": 7.0552}, {"x": -2.6715, "y": 6.3663, "z": 7.0539}, {"x": -2.6737, "y": 6.3657, "z": 7.0538}, {"x": -2.6725, "y": 6.3670, "z": 7.0523}, {"x": -2.6735, "y": 6.3696, "z": 7.0527}, {"x": -2.6750, "y": 6.3688, "z": 7.0542}]}

```

Setelah melakukan consume, maka program consumer akan mengirimkan pesan menuju MySQL. Hasilnya adalah sebagai berikut:

The screenshot shows the MySQL Workbench interface with a query results grid. The query executed is:

```
SELECT * FROM earthquake.store_id;
```

The results grid displays the following data:

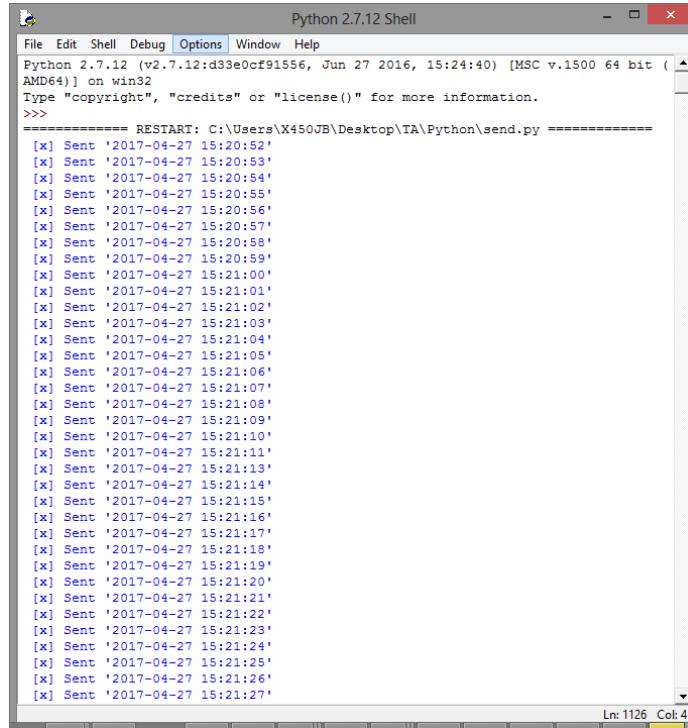
point_time	time_zone_id	interval_id	client_id	latitude_id	longitude_id
2017-04-26 14:29:02	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:04	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:05	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:06	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:07	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:08	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:09	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:10	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:11	Asia/Jakarta	1000	ECN-1	-6.89	107.61
2017-04-26 14:29:12	Asia/Jakarta	1000	ECN-1	-6.89	107.61
NULL	NULL	NULL	NULL	NULL	NULL

Data yang tersimpan ini akan digunakan untuk pengolahan data gempa.

2.2.3 Pengujian Latency Jaringan Sistem

Namun selain kemampuan untuk dapat mengirimkan pesan, sistem ini juga harus handal untuk dapat berjalan selama mungkin. Oleh karena itu, keaslian data secara aktual juga harus

diperhitungkan. Pengujian dilakukan dengan mengirimkan sebuah pesan Timestamp dari sebuah producer kemudian mencocokkan Timestamp tsb dengan Timestamp consumer. Hasil pengiriman Timestamp producer dengan menggunakan Python adalah sebagai berikut:

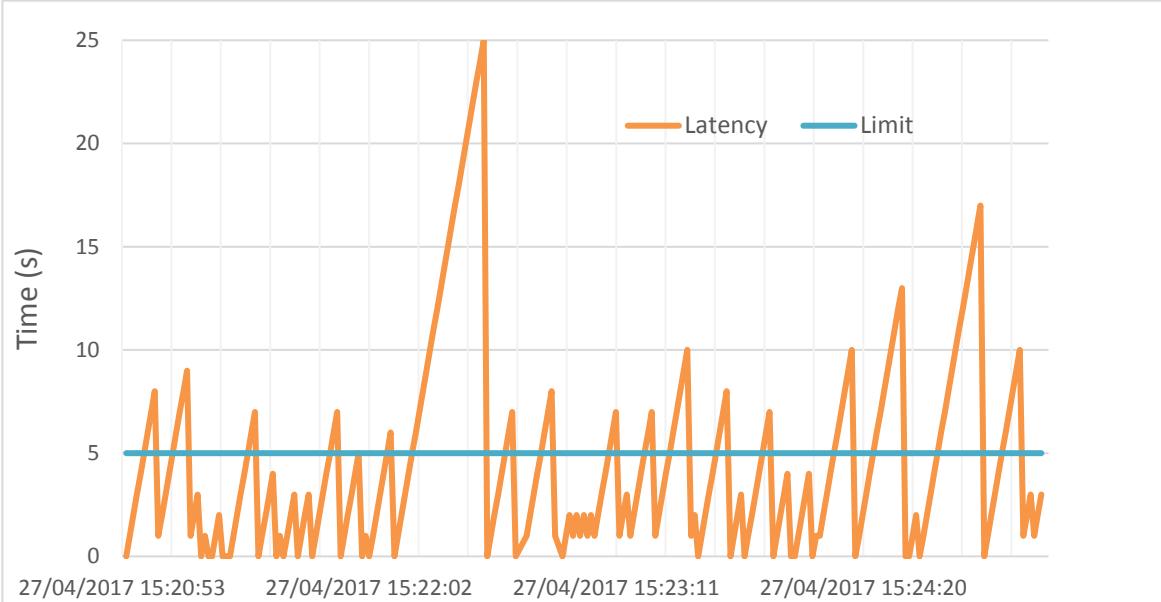


```

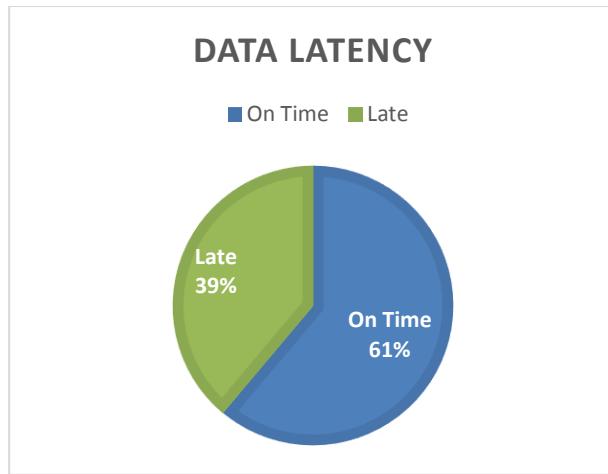
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:24:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\X450JB\Desktop\TA\Python\send.py =====
[x] Sent '2017-04-27 15:20:52'
[x] Sent '2017-04-27 15:20:53'
[x] Sent '2017-04-27 15:20:54'
[x] Sent '2017-04-27 15:20:55'
[x] Sent '2017-04-27 15:20:56'
[x] Sent '2017-04-27 15:20:57'
[x] Sent '2017-04-27 15:20:58'
[x] Sent '2017-04-27 15:20:59'
[x] Sent '2017-04-27 15:21:00'
[x] Sent '2017-04-27 15:21:01'
[x] Sent '2017-04-27 15:21:02'
[x] Sent '2017-04-27 15:21:03'
[x] Sent '2017-04-27 15:21:04'
[x] Sent '2017-04-27 15:21:05'
[x] Sent '2017-04-27 15:21:06'
[x] Sent '2017-04-27 15:21:07'
[x] Sent '2017-04-27 15:21:08'
[x] Sent '2017-04-27 15:21:09'
[x] Sent '2017-04-27 15:21:10'
[x] Sent '2017-04-27 15:21:11'
[x] Sent '2017-04-27 15:21:13'
[x] Sent '2017-04-27 15:21:14'
[x] Sent '2017-04-27 15:21:15'
[x] Sent '2017-04-27 15:21:16'
[x] Sent '2017-04-27 15:21:17'
[x] Sent '2017-04-27 15:21:18'
[x] Sent '2017-04-27 15:21:19'
[x] Sent '2017-04-27 15:21:20'
[x] Sent '2017-04-27 15:21:21'
[x] Sent '2017-04-27 15:21:22'
[x] Sent '2017-04-27 15:21:23'
[x] Sent '2017-04-27 15:21:24'
[x] Sent '2017-04-27 15:21:25'
[x] Sent '2017-04-27 15:21:26'
[x] Sent '2017-04-27 15:21:27'

```

Kemudian pesan di dalam Messaging Server di consume oleh console C# dan digabungkan dengan Timestamp consumer. Kedua time stamp ini kemudian disimpan ke dalam Database MySQL dan di export datanya untuk pengolahan lebih lanjut. Hasil pengolahannya adalah sebagai berikut:



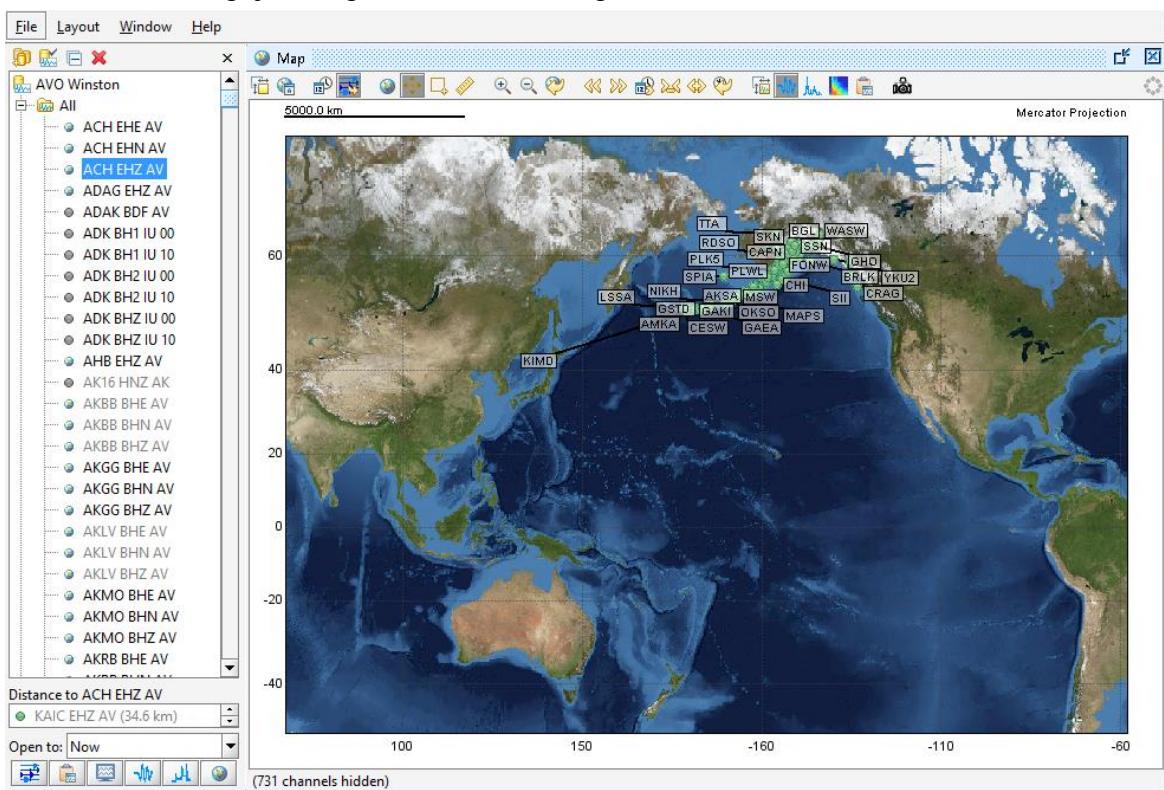
Artinya terdapat beberapa buah data yang tidak real time karena terlambat untuk sampai ke pengolahan server. Setelah diolah datanya, didapatkan hasil sebagai berikut:



Terdapat peluang sebesar 61% bahwa data yang dikirimkan adalah data real-time ketika melalui sistem jaringan.

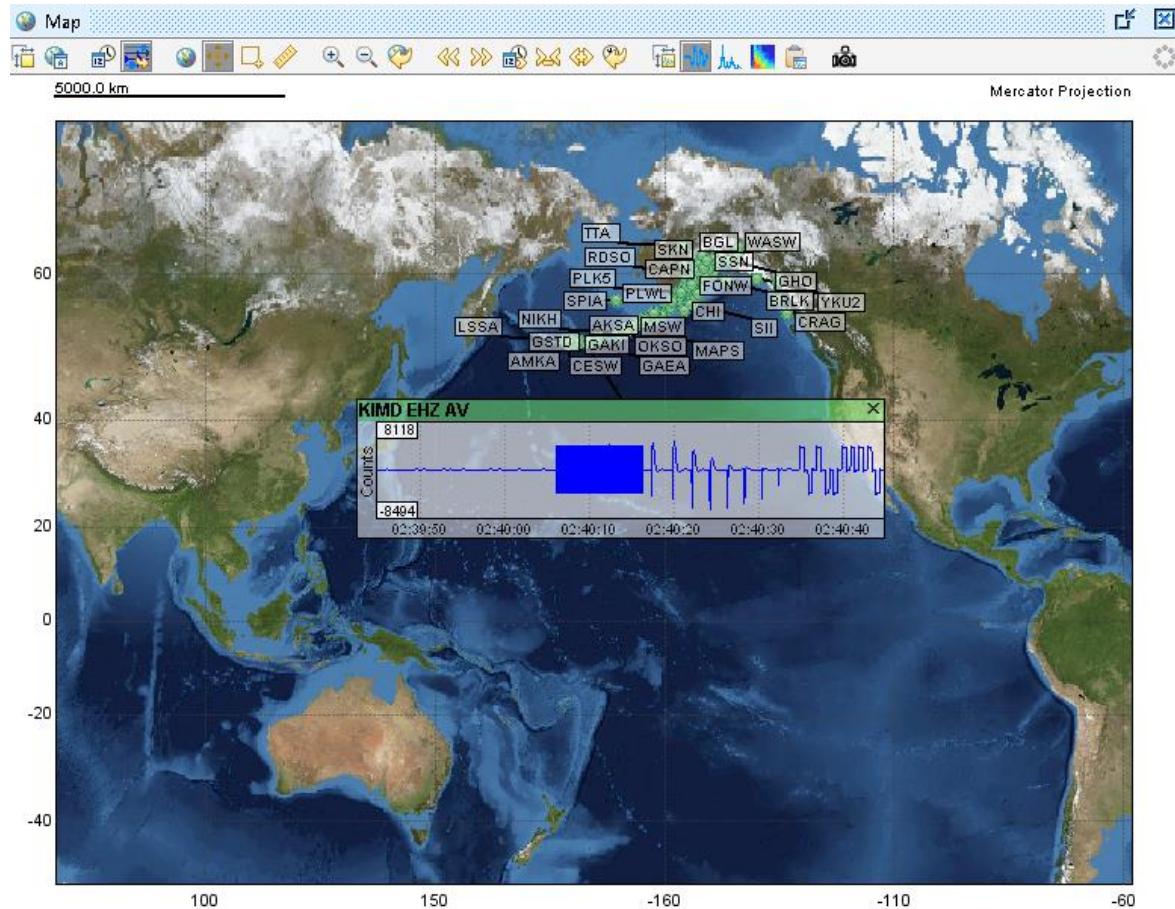
2.3 Pengujian Algoritma Analisis Gempa

2.3.1 Pengujian Algoritma Umum Program



Algoritma umum yang dimaksud disini adalah fungsionalitas dasar dari program analisis yang dibuat, yaitu seperti menampilkan peta yang disediakan oleh NASA, koneksi ke jaringan sensor gempa yang dimiliki USGS, serta berfungsinya semua menu dan toolbar yang disediakan pada tampilan. Dari gambar diatas dapat dilihat beberapa fungsi yang sudah berjalan dengan cukup baik. Pada sebelah kiri terdapat daftar sensor yang bisa diakses oleh program. Sensor dengan lingkaran hijau adalah sensor yang bisa diakses saat itu juga, sedangkan sensor dengan lingkaran hijau pucat adalah sensor yang aktif namun sedang

mengalami kesulitan untuk mengirim data. Sensor yang tidak aktif ditandai dengan lingkaran berwarna abu-abu. Pada sebelah kanan terdapat peta dengan penanda GPS untuk masing-masing sensor yang aktif. Beberapa tombol penting yang bisa digunakan untuk pengaturan tampilan antara lain yaitu pembesaran/pengecilan, koneksi ke server baru, tampilan *waveform*, serta mode kiosk.

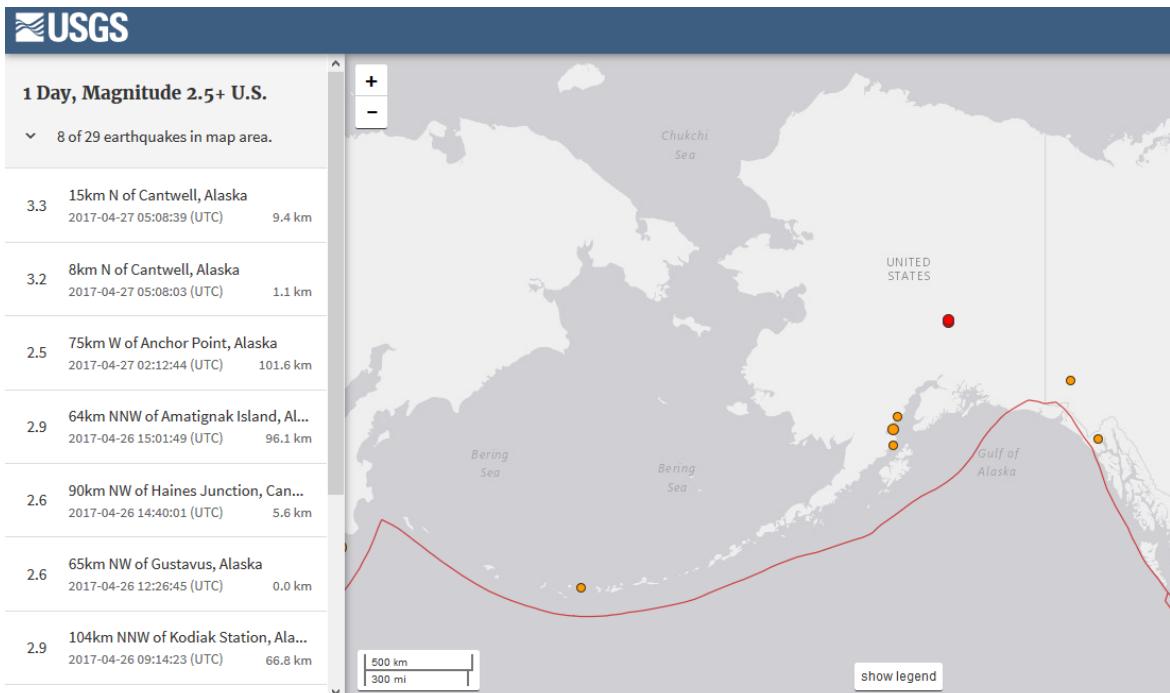


Tampilan diatas menunjukkan pembacaan sensor yang didapat ketika salah satu penanda GPS pada peta diklik. Pembacaan tersebut adalah data yang didapat sejak satu menit terakhir, dihitung dari data terakhir yang didapat dari sensor. Pembacaan data dari semua sensor belum berlangsung secara *real-time*, karena ada *latency* dari pengiriman data oleh sensor hingga data tiba di database dan bisa diproses. Data *latency* dari masing-masing sensor dapat dilihat pada tabel berikut.

Lokasi Sensor	Jarak Sensor ke Pemancar GSM	Latency (Rata-rata)
KAKN BHV	3.2 km	1.2s
GSTD EHN	5.0 km	1.1s
KABU BHE	7.2 km	1.9s
KCE EHZ	9.1 km	1.7s
ANCK EHZ	9.9 km	4.3s
KCG EHE	16.5 km	3.2s
KAIC EHZ	34.6 km	3.6s

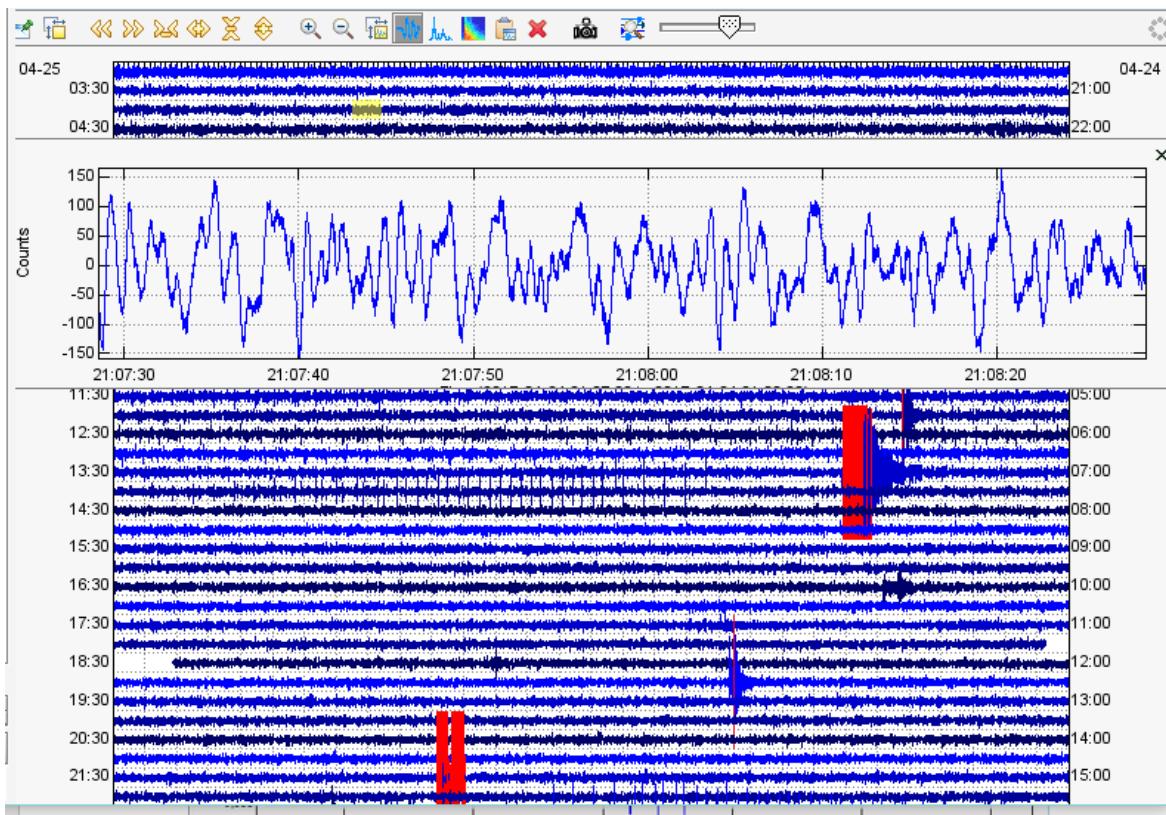
Data tersebut diambil dari jaringan sensor yang dimiliki oleh USGS. Namun, kami berekspektasi bahwa *latency* pada jaringan kami hampir sama karena desainnya serupa. Jaringan yang digunakan adalah GSM. Dengan data seperti diatas, GSM dirasa sudah cukup cepat untuk mentransmisikan data dari sensor ke server.

2.3.2 Pengujian Algoritma *Counting*



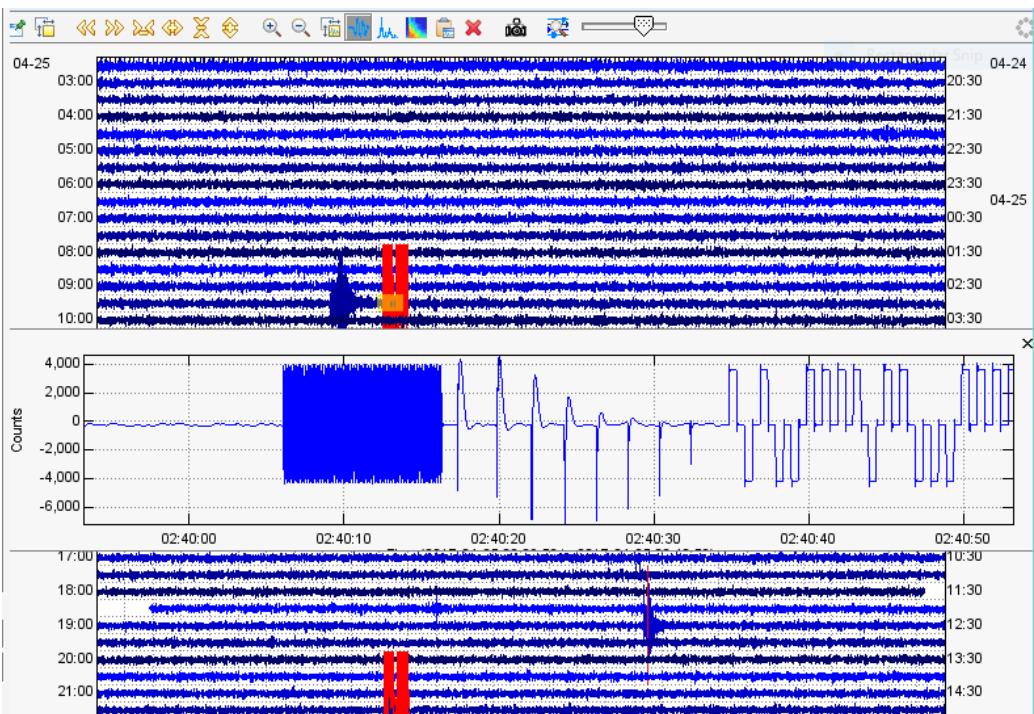
Untuk melakukan pengujian algoritma analisis data seismik kami, digunakan cara sebagai berikut. Kami melakukan analisis dengan menerapkan algoritma kami pada data dari jaringan sensor gempa yang dimiliki USGS. Jaringan sensor tersebut terletak di Alaska, Amerika Serikat. Untuk membuktikan apakah kesimpulan dari hasil analisis kami benar atau tidak, kami membandingkannya dengan hasil analisis yang dilakukan oleh USGS sendiri. Hasil tersebut dapat dilihat di *website* USGS, dimana dapat dilihat gempa terakhir yang terjadi di suatu negara bagian di Amerika Serikat. Data yang ditampilkan seperti gambar diatas. Data tersebut berupa *magnitude*, jarak pusat gempa dari tempat terdekat yang berpenghuni, kedalaman gempa dari permukaan laut, serta waktu terjadinya gempa tersebut. Kami memilih untuk membandingkan dengan jaringan gempa di Alaska, karena di daerah tersebut cukup sering terjadi gempa.

Implementasi algoritma kami menghitung *magnitude* dari gempa yang terjadi, frekuensi getaran, *count* dari getaran, serta spektrum frekuensi dari getaran, yang nantinya akan digunakan untuk menentukan lokasi gempa. Namun, hingga dokumen ini ditulis, implementasi dari algoritma untuk menentukan lokasi gempa belum selesai diimplementasikan. Maka itu, kami hanya membandingkan *magnitude* dan frekuensi dari gempa hasil analisis kami sendiri dengan hasil yang diperoleh dari *website* USGS. Mengenai algoritma *count*, fungsi dari algoritma ini adalah menghitung jumlah perubahan arah *jerk*, atau turunan dari akselerasi per jam. Nilai *count* yang kurang dari 100 bisa dianggap sebagai keadaan normal. Apabila nilai *count* melebihi 3600, atau 1 Hz, maka bisa dianggap bahwa sensor tersebut mendeteksi ada gempa yang terjadi.

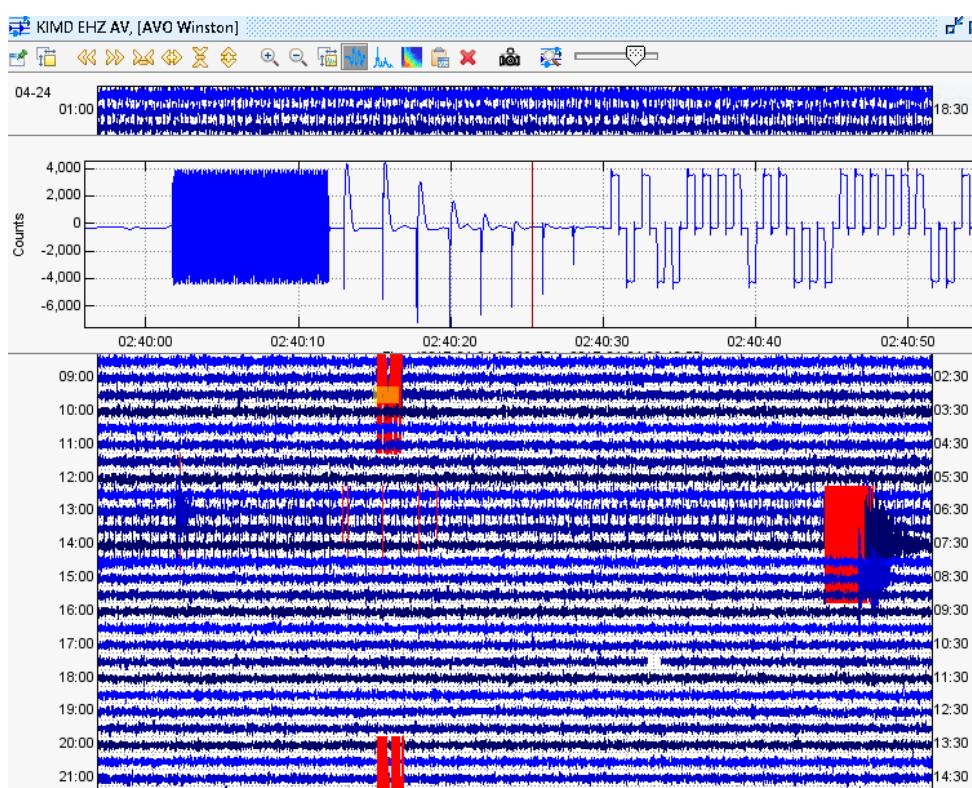


Gambar diatas adalah penerapan algoritma *count* pada data ketika keadaan normal. Bisa dilihat bahwa nilai *count* tidak ada yang melebihi 200. Dengan kata lain, ada getaran namun dengan frekuensi yang sangat rendah. Getaran ini dianggap sebagai *noise*, karena sensor tidak didesain untuk mendeteksi getaran dengan frekuensi dibawah 1 Hz. Bagian yang *highlight* dengan warna kuning adalah bagian dari rekaman sensor gempa yang menjadi fokus. Bagian yang bisa menjadi fokus dalam satu waktu sepanjang sekitar 1 menit.

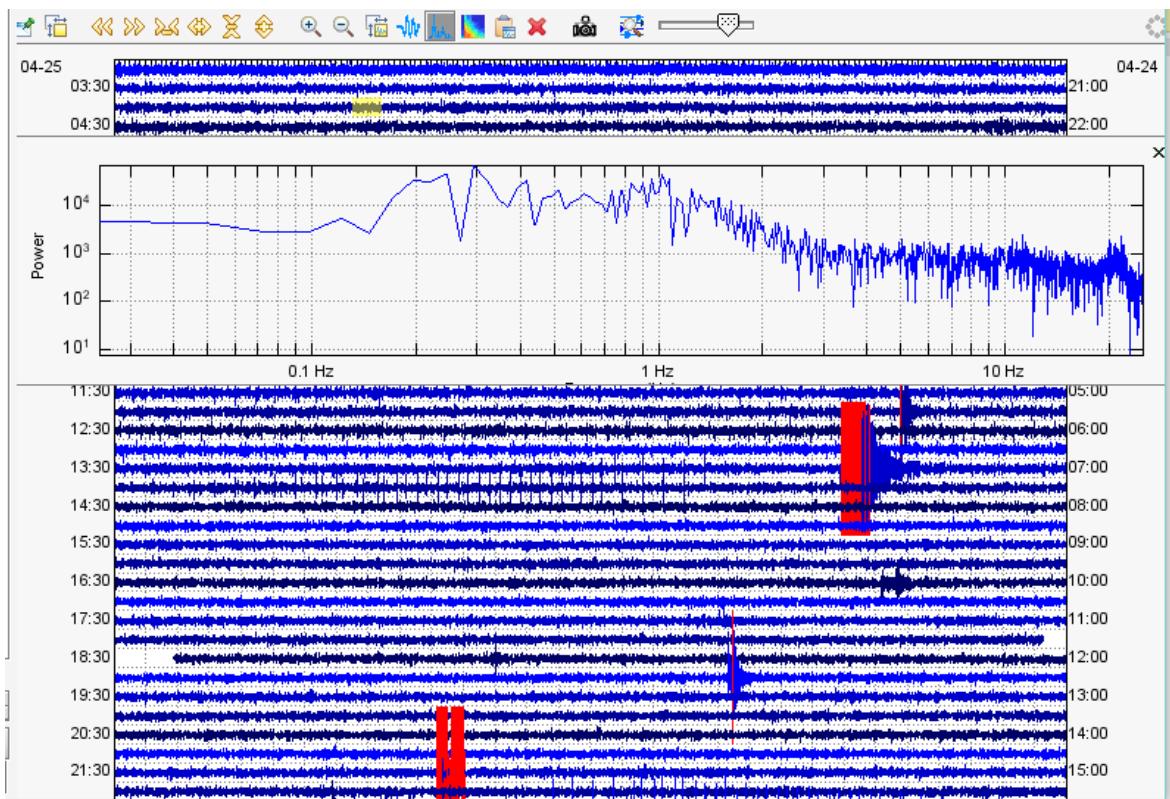
Warna merah tersebut menunjukkan adanya getaran yang dideteksi diatas 1 Hz. Getaran seperti ini mengindikasikan adanya gempa. Namun, bisa juga ada getaran lokal yang dialami oleh salah satu sensor, seperti kendaraan yang lewat atau semacamnya. Maka itu, apabila data gempa diperoleh pada satu sensor saja, belum dianggap terjadi gempa. Dibutuhkan minimal tiga sensor yang mendeteksi gempa pada waktu yang berdekatan untuk menganggap bahwa gempa telah terjadi. Angka tiga ini dipilih karena dibutuhkan minimal tiga sensor untuk melakukan analisis mengenai lokasi gempa. Contoh dari getaran yang tidak dianggap gempa adalah warna merah yang terjadi pada jam 07.00 di gambar diatas, serta warna merah saat jam 15.00



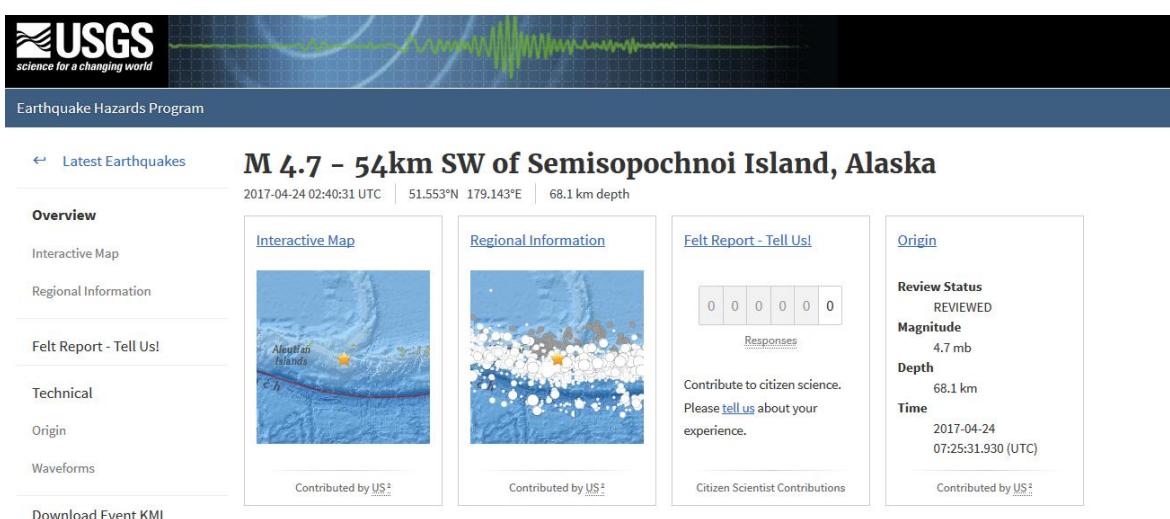
Gambar diatas adalah contoh dari getaran yang diambil saat gempa sungguhan terjadi. Data getaran tersebut juga dialami oleh sensor-sensor lain, dengan adanya perbedaan pada waktu terjadinya getaran tersebut. Perbedaan waktu ini terjadi karena getaran gempa perlu waktu untuk merambat ke sensor yang letaknya cukup jauh. Bila bagian yang berwarna merah difokuskan, maka bisa dilihat bahwa saat terjadi gempa, nilai *count* berubah dengan sangat cepat dan signifikan. Hal ini artinya terjadi getaran dengan frekuensi yang dianggap cukup tinggi untuk sebuah gempa tektonik. Pada gambar dibawah, getaran yang sama terkam di sensor lain, dengan perbedaan waktu yang kecil.



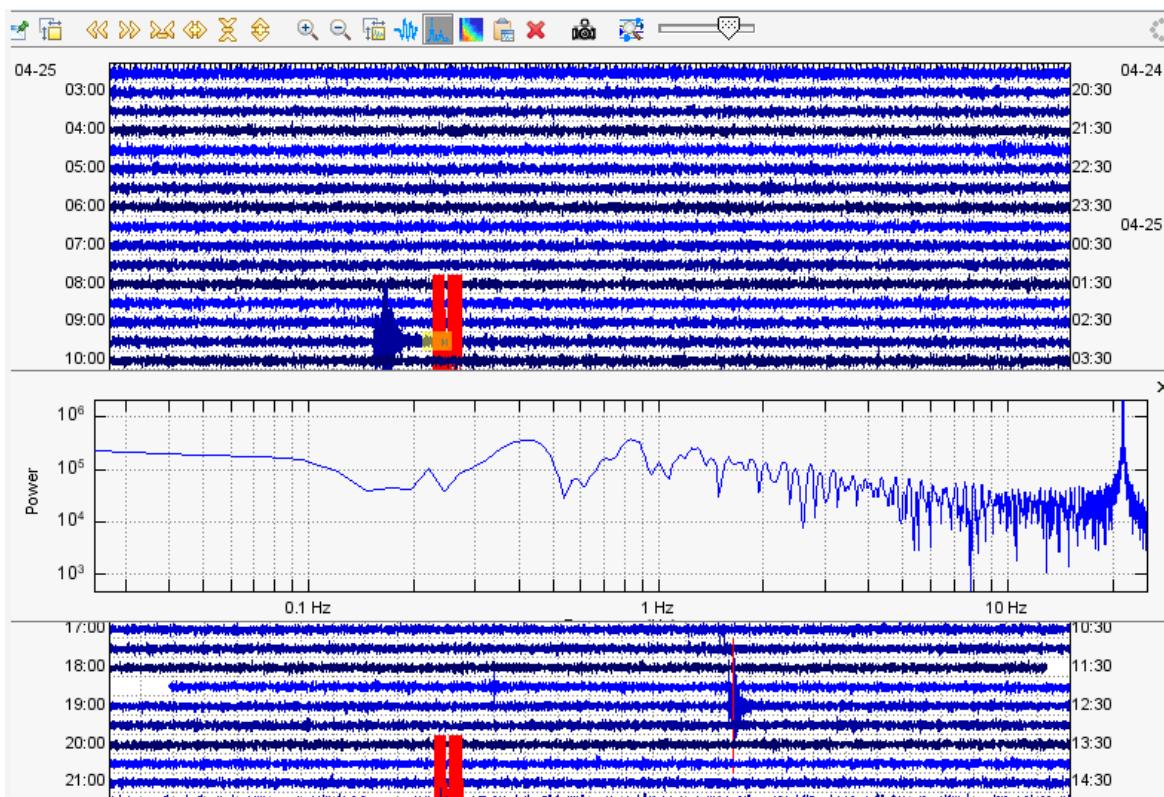
2.3.3 Pengujian Algoritma Magnitude



Bagian ini akan menjelaskan pengujian algoritma untuk menghitung *magnitude* dari gempa. *Magnitude* disini didasarkan pada *moment magnitude*, atau kekuatan gempa yang diukur berdasarkan jarak perpindahan lempeng tektonik, dan gaya yang dibutuhkan untuk melakukan perpindahan tersebut. Pengujian dengan metode ini dirasa lebih baik dibandingkan dengan skala *Richter*, karena skala momen lebih akurat untuk gempa dengan kekuatan besar. Pada keadaan normal, analisis menghasilkan data seperti gambar diatas. Diatas frekuensi 1-2 Hz, *magnitude* yang terukur selalu kurang dari 3, yaitu batas ketelitian sensor. Karena itu, data dengan *magnitude* dibawah 3 dianggap bukan gempa.



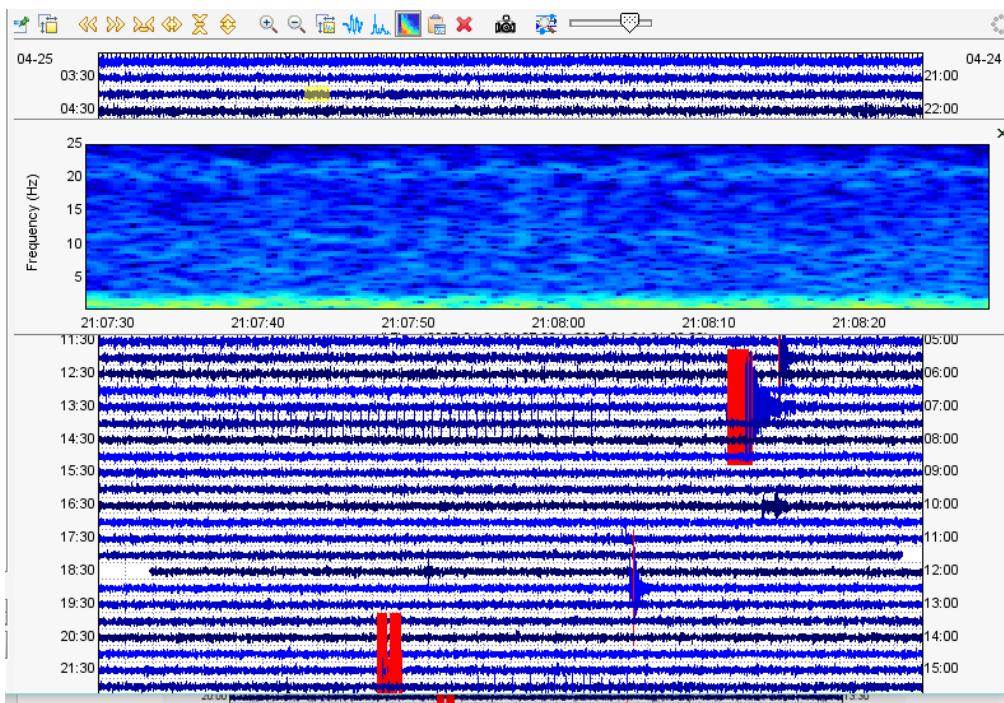
Pada gambar diatas, terdapat data gempa yang terjadi di sekitar Alaska, pada jam 02.40 dengan kekuatan 4,7. Gempa yang sama terdeteksi pada sensor seperti yang terlihat pada gambar dibawah. Pada analisis ini, terlihat ada lonjakan pada *magnitude* gempa dengan frekuensi sekitar 11 Hz. Puncaknya sekitar *magnitude* 6, namun rata-rata dari lonjakan tersebut sekitar 5, atau tepatnya 4,93. Hal ini hampir sama dengan kesimpulan yang diperoleh USGS.



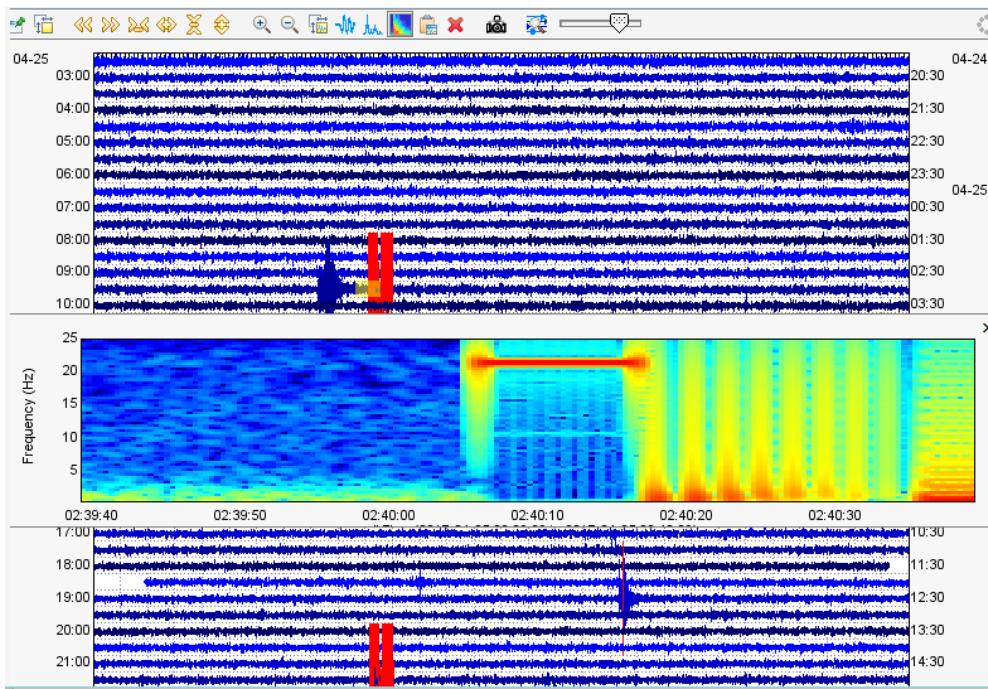
Tabel dibawah menunjukkan beberapa gempa yang terjadi sebelum dokumen ini ditulis, data yang berkaitan, dan hasil analisis menggunakan algoritma kami. Dari tabel tersebut dapat disimpulkan bahwa algoritma sudah cukup akurat untuk gempa diatas skala 4, namun dibawah itu hasil analisis tidak bisa diandalkan.

No.	Waktu	Lokasi	Kedalaman	Magnitude USGS	Magnitude Algoritma
1	2017-04-26 07:02:28 (UTC)	53.828°N 160.645°E	39.8 km	4.9	4.8765
2	2017-04-26 15:01:49 (UTC)	51.817°N 179.367°W	96.1 km	2.9	3.5638
3	2017-04-26 09:14:23 (UTC)	58.548°N 153.567°W	66.8 km	2.9	3.3653
4	2017-04-27 02:12:44 (UTC)	59.750°N 153.179°W	101.6 km	2.5	3.3673
5	2017-04-27 05:08:39 (UTC)	63.532°N 148.976°W	9.4 km	3.3	3.6563

2.3.4 Pengujian Algoritma Spektrum Frekuensi



Algoritma spektrum frekuensi menggunakan transformasi Fourier dengan metode Fast Fourier Transform (FFT) untuk menentukan spektrum frekuensi dari gempa pada waktu tertentu. Spektrum frekuensi ini nantinya akan digunakan untuk menentukan lokasi dari gempa. Namun hingga dokumen ini ditulis, algoritma lokasi belum selesai diimplementasikan sehingga belum bisa diuji.



LAMPIRAN

Tabel perhitungan lama proses algoritma dengan kondisi tanpa menunggu queue RabbitMQ

No	Process Time (ms)		
	IMU Processing	GPS Processing	MQTT Publish Processing
1	1	0	256
2	9	0	212
3	1	1	212
4	18	3	228
5	1	0	210
6	17	3	236
7	2	0	212
8	1	0	221
9	17	3	211
10	16	0	248
11	1	0	217
12	16	4	229
13	1	0	208
14	17	2	220
15	1	0	208
16	2	0	221
17	17	3	208
18	1	0	209
19	1	0	246
20	2	0	214
21	17	2	218
22	2	0	208
23	1	0	252
24	18	3	206
25	1	1	209
26	17	2	231
27	1	0	245
28	1	0	215
29	18	3	207
30	1	0	234
31	17	2	223
32	1	0	217
33	16	2	236
34	1	0	217
35	16	2	236

36	1	0	224
37	18	2	233
38	1	1	224
39	17	2	228
40	18	2	275
41	1	0	269
42	1	0	254
43	1	0	231
44	1	0	227
45	18	2	235
46	2	0	214
47	17	3	235
48	1	0	215
49	1	0	271
50	16	2	251
51	1	0	214
52	17	2	274
53	1	1	221
54	18	2	243
55	2	0	238
56	1	0	276
57	17	3	227
58	1	0	276
59	17	3	274
60	1	0	322
Average	7,666666667	1,016666667	232,1833333
Maximum	18	4	322
Minimum	1	0	206

Source Code GUI ECN Data Logger

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using ClosedXML.Excel;

namespace ECN_Data_Logger
{
    public partial class FormECNDataLogger : Form
    {
        static string DataXYZ;
        public float[] Data = new float[3];
        public bool record = false;
        DataTable table = new DataTable();
        public float[,] Sample = new float[120,3];
        int i = 0, N=0;

        public FormECNDataLogger()
        {
            InitializeComponent();
            GetPortName();
            table.Columns.Add("Acc X", typeof(string));
            table.Columns.Add("Acc Y", typeof(string));
            table.Columns.Add("Acc Z", typeof(string));
        }

        void GetPortName()
        {
            String[] portname = SerialPort.GetPortNames();
            PortNameBox.Items.AddRange(portname);
        }

        private void OpenPortButton_Click(object sender, EventArgs e)
        {
            try
            {
                if(PortNameBox.Text == "" || BaudRateBox.Text == "")
                {
                    MessageBox.Show("Please select port settings !!!","Error");
                }
                else
                {
                    SerialPort.PortName = PortNameBox.Text;
                    SerialPort.BaudRate = Convert.ToInt32(BaudRateBox.Text);
                    //SerialPort.DataReceived += new
                    SerialDataReceivedEventHandler(DataReceivedHandler);
                    SerialPort.Open();
                    StatusBar.Value = 100;
                }
            }
        }

        void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
        {
            if(SerialPort.BytesToRead > 0)
            {
                byte[] buffer = new byte[SerialPort.BytesToRead];
                SerialPort.Read(buffer, 0, SerialPort.BytesToRead);
                DataXYZ = BitConverter.ToString(buffer).Replace("-", " ");
                Data[0] = Convert.ToSingle(DataXYZ.Substring(0, 4));
                Data[1] = Convert.ToSingle(DataXYZ.Substring(5, 4));
                Data[2] = Convert.ToSingle(DataXYZ.Substring(10, 4));
                table.Rows.Add(Data[0], Data[1], Data[2]);
                Sample[i, 0] = Data[0];
                Sample[i, 1] = Data[1];
                Sample[i, 2] = Data[2];
                i++;
                if(i == 120)
                {
                    i = 0;
                    N++;
                    if(N == 10)
                    {
                        N = 0;
                        table.Rows.Add(Data[0], Data[1], Data[2]);
                    }
                }
            }
        }
    }
}
```

```

        ClosePortButton.Enabled = true;
        Box.Enabled = true;
        StartButton.Enabled = true;
        StopButton.Enabled = false;

    }

}

catch(UnauthorizedAccessException)
{
    MessageBox.Show("Unauthorized Access !!!", "Error");
}
}

private void ClosePortButton_Click(object sender, EventArgs e)
{
    timer1.Stop();
    SerialPort.Close();
    StatusBar.Value = 0;
    Box.Enabled = false;
    ClosePortButton.Enabled = false;
    StopButton.Enabled = false;
    StartButton.Enabled = true;

}
/*
private static void DataReceivedHandler( object sender,
SerialDataReceivedEventArgs e)
{
    SerialPort sp = (SerialPort)sender;
    string indata = sp.ReadExisting();
    DataXYZ = indata;
}
*/
}

private void timer1_Tick(object sender, EventArgs e)
{
/*
try
{*/
    DataXYZ = SerialPort.ReadLine();
    SerialDataBox.AppendText(DataXYZ);
    SerialDataBox.AppendText("\n");
    char[] delimiterChars = {'\t'};
    String[] XYZ = DataXYZ.Split(delimiterChars);
    //Data[0] = Convert.ToSingle(XYZ[0]);
    //Data[1] = Convert.ToSingle(XYZ[1]);
    //Data[2] = Convert.ToSingle(XYZ[2]);
    XYZ[0] = XYZ[0].Replace(".", ",");
    XYZ[1] = XYZ[1].Replace(".", ",");
    XYZ[2] = XYZ[2].Replace(".", ",");

    if (record)
    {
        table.Rows.Add(XYZ[0],XYZ[1],XYZ[2]);
        /*
        N++;
        Sample[i,0] = Data[0];
        Sample[i,1] = Data[1];
        Sample[i,2] = Data[2];
    }
}

```

```

        i++;

        if (i == 120)
            i = 0;
        */
    }

    /*
}
catch (Exception)
{
    MessageBox.Show("Exception !!!","Error");
}
*/
}

private void StartButton_Click(object sender, EventArgs e)
{
    timer1.Start();
    StartButton.Enabled = false;
    StopButton.Enabled = true;
}

private void StopButton_Click(object sender, EventArgs e)
{
    timer1.Stop();
    StopButton.Enabled = false;
    StartButton.Enabled = true;
}

private void ActivityBox_TextChanged(object sender, EventArgs e)
{
}

private void ClearButton_Click(object sender, EventArgs e)
{
    SerialDataBox.Clear();
}

private void RecordButton_Click(object sender, EventArgs e)
{
    StopRecordButton.Enabled = true;
    RecordButton.Enabled = false;
    record = true;
    i = 0;
    //table.Rows.Add("Amplitude", AmplitudeBox.Text,"G");
    //table.Rows.Add("Frequency", FrequencyBox.Text,"Hz");

}

private void StopRecordButton_Click(object sender, EventArgs e)
{
    StopRecordButton.Enabled = false;
    RecordButton.Enabled = true;
    record = false;
    /*for(i=0;i<120;i++)
    {
        table.Rows.Add(Sample[i,0], Sample[i,1], Sample[i,2]);
    }*/
}

```

```

        DataGridView.DataSource = table;
    }

    private void ClearDataButton_Click(object sender, EventArgs e)
    {
        table.Clear();
        DataGridView.DataSource = table;
        i = 0;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            SaveFileDialog sfd = new SaveFileDialog();
            sfd.Filter = "Excel Documents (*.xlsx)|*.xlsx";
            String name = comboBox1.SelectedItem.ToString() + " " +
AmplitudeBox.SelectedItem.ToString() + "G " +
FrequencyBox.SelectedItem.ToString() + "Hz";
            sfd.FileName = "Test MPU9255 " + name + ".xlsx";
            if (sfd.ShowDialog() == DialogResult.OK)
            {
                var workbook = new XLWorkbook();

                workbook.Worksheets.Add(table, "Test MPU9255");
                workbook.SaveAs(sfd.FileName);
                table.Clear();
                MessageBox.Show("Save File Success !", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);

            }
        }
        catch
        {
            MessageBox.Show("Save File Failed", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```