**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 |  | |
|  |  | |
| Tanggal Penerbitan | 2 December 2016 | |
|  |  | |
| Unit Penerbit | Prodi Teknik Elektro – ITB | |
|  |  | |
| Jumlah Halaman | 17 | (termasuk lembar sampul ini) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Pemeriksaan dan Persetujuan | | | | |
| Ditulis | Nama | **Christoporus Deo Putratama** | Jabatan | Anggota |
| Oleh | Tanggal | Desember 2016 | Tanda Tangan |  |
|  | Nama | **Kevin Shidqi** | Jabatan | Anggota |
|  | Tanggal | Desember 2016 | Tanda Tangan |  |
|  | Nama | **Bramantio Yuwono** | Jabatan | Anggota |
|  | Tanggal | Desember 2016 | Tanda Tangan |  |
|  |  |  |  |  |
| Diperiksa | Nama | **Dr. techn. Ary Setijadi**  **Prihatmanto** | Jabatan | Dosen Pembimbing |
| Oleh | Tanggal | Desember 2016 | Tanda Tangan |  |
|  |  |  |  |  |
| Disetujui | Nama | **Dr. techn. Ary Setijadi**  **Prihatmanto** | Jabatan | Dosen Pembimbing |
| Oleh | Tanggal | Desember 2016 | Tanda Tangan |  |
|  |  |  |  |  |

DAFTAR ISI

[DAFTAR ISI 3](#_Toc462040353)

[Catatan Sejarah Perbaikan Dokumen 4](#_Toc462040354)

[Proposal Proyek Pengembangan SISTEM DETEKTOR GEMPA DAN TSUNAMI DECISION SUPPORT SYSTEM 5](#_Toc462040355)

[1 Pengantar 5](#_Toc462040356)

[1.1 RINGKASAN ISI DOKUMEN 5](#_Toc462040357)

[1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen 5](#_Toc462040358)

[1.3 REFERENSI 5](#_Toc462040359)

[1.4 DAFTAR SINGKATAN 6](#_Toc462040360)

[2 pERANCANGAN 6](#_Toc462040361)

[2.1 Desain Sistem 6](#_Toc462040362)

[2.2 Desain Komunikasi 7](#_Toc462040363)

[2.3 Desain Hardware 8](#_Toc462040364)

[2.4 Desain Software 12](#_Toc462040365)

[2.5 Desain Pemodelan 15](#_Toc462040366)

[3 Lampiran 17](#_Toc462040367)

Catatan Sejarah Perbaikan Dokumen

|  |  |
| --- | --- |
| Versi, Tgl, Oleh | Perbaikan |
|  |  |
|  |  |
|  |  |
|  |  |

Proposal Proyek Pengembangan Rancang Bangun *Flapping Wing Microaerial Vehicle* untuk Misi Monitoring dengan Kamera

# Pengantar

## 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

## 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.

## REFERENSI

[1] Croon, G.C.H.E de. The DelFly :*Design, Aerodynamics, and Artificial Intelligence of a Flapping Wing Robot*. Springer: (2016)

[2] DiLeo, Christopher dan Xinyan Deng. *Design and Experiments of a Dragonfly-Inspired Robot*, 2009. <https://engineering.purdue.edu/-xdeng/AR09.pdf> ,6 September 2016, 18.58 WIB.

[3] Diardano R., Oki N. dan Dwicky F.S., Rancang Bangun *Flapping Wings Microaerial Vehicle* : Sistem Kendali, Sensor dan Telemetri, Proposal Tugas Akhir, Institut Teknologi Bandung, 2016.

## DAFTAR SINGKATAN

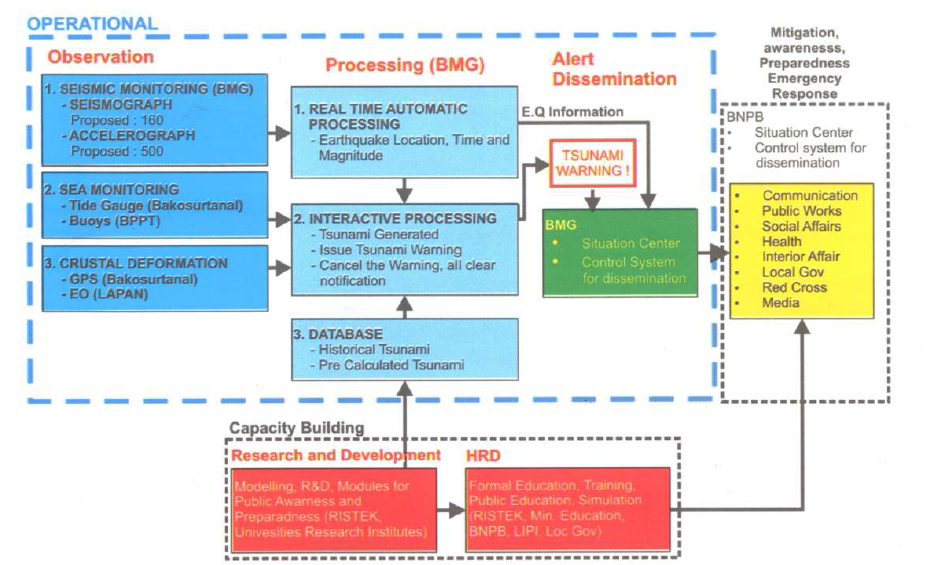
| Singkatan | Arti |
| --- | --- |
| MAV | *Micro Aerial Vehicle* |
| GCS | *Ground Control Station* |
| GPS | *Global Positioning System* |
| IMU | *Inertial Measurement Unit* |
| UAV | *Unmanned Aerial Vehicle* |
| CPU | *Central Processing Unit* |
| GUI | *Graphical User Interface* |
| 3D | 3 Dimensi |

# PERANCANGAN

## Desain Sistem

## Desain Sistem

*Earthquake Cathcer Network* adalah sistem pendeteksi 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 mendeteksi suatu kejadian gempa, menentukan lokasi dan kedalaman pusat gempa, menentukan magnituda 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.





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

1. **Komponen Operasional**

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

1. **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

1. **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.

1. **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.

1. **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.



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 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.

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 peformasi 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.

## Desain Komunikasi

### **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:

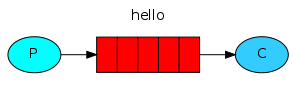
|  |
| --- |
| Tsunami Warning in BENGKULU, Eq Mag:7.0RS, 09-Dec-09 15:52:59 UTC, Loc:4.64S/101.11E,Dep:10km::BMKG |

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:



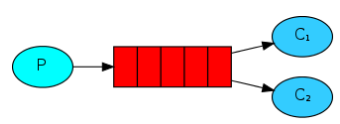
Queue

Consuming

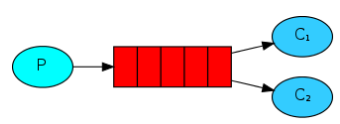
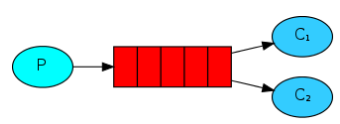
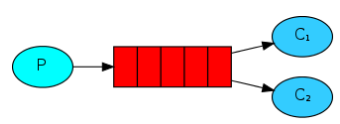
Producing

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.

[Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]



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



## Desain Hardware

### **Sensor**

### **Power System**

*.*Pada sensor yang didesain, diperlukan mekanisme *self-powering* yang dilakukan oleh sensor tersebut. Pada sensor ECN ini, digunakan solar panel sebagai sumber daya sensor. Solar panel harus memiliki spesifikasi yang tepat agar sistem daya sensor ini dapat men-*charge* sensor pada siang hari dan disimpan ke baterai sehingga ketika malam hari, baterai memiliki kapasitas yang cukup untuk mensuplai daya ke sensor pada malam hari. Oleh karena itu diperlukan suatu sistem *charge controller* yang akan mengatur proses *charging* yang dilakukan solar panel ke baterai. Diagram *power system* dapat dilihat pada gambar berikut.

Sensor

Charge Controller

Solar Panel

Battery

**Gambar Diagram Blok Power System**

Fungsi *charge controller* adalah untuk mengatur tingkat *charging* yang dilakukan oleh solar panel ke baterai agar tidak terjadi *overcharge* pada baterai yang dapat menyebabkan baterai dapat mengalami kerusakan. Agar *power system* dapat menyuplai daya sensor selama 24 jam, solar panel dan baterai harus memiliki spesifikasi yang sesuai. Dengan daya yang diperlukan sensor sebesar 5 watt dan kemampuan solar panel yang dapat menyuplai daya maksimum selama 7 jam pada kondisi cuaca cerah, dapat diperoleh spesifikasi kapasitas dan daya solar panel dan baterai yang diperlukan.

Pada malam hari :

Pada siang hari :

Sehingga diperoleh baterai dan solar panel yang sesuai. Solar panel harus memiliki kemampuan suplai daya minimum sebesar 13.5 W dan dengan kapasitas minimum 18900 mAh. Baterai harus memiliki kapasitas minimum sebesar 12000 mAh.

Agar menghindari *overcharge* diperlukan charge controller yang dapat mengatur tingkat *charging* yang dilakukan solar panel ke baterai. Diagram blok *charge controller* dapat dilihat pada gambar berikut.

P-MOS

P-MOS

Sensor

Battery

Solar Panel

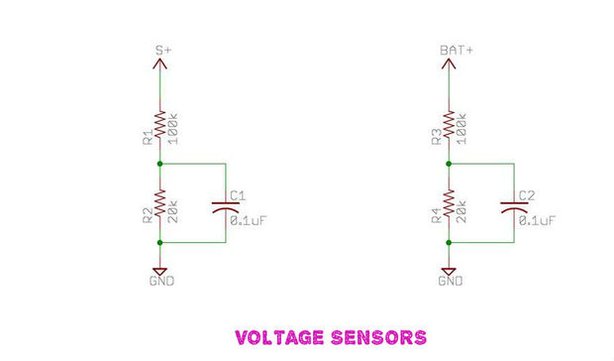
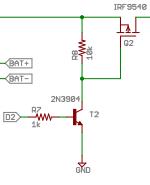
Voltage Sensor

Voltage Sensor

MCU

**Gambar Diagram Charge Controller**

Pada charge controller ini digunakan dua buah P-MOS untuk melakukan switching tegangan dari solar panel ke baterai ataupun baterai ke sensor. Selain itu, digunakan juga sensor tegangan untuk mengukur tegangan baterai dan solar panel. MCU yang digunakan adalah arduino nano. Pemilihan MCU ini berdasarkan oleh harga arduino nano yang murah serta memiliki semua fitur yang dibutuhkan dalam *charge controller* ini. Implementasi rangkaian sensor tegangan dan rangkaian switching P-MOS dapat dilihat pada gambar berikut.

****

**Gambar Implementasi Rangkaian Sensor Tegangan dan Switching P-MOS**

Dalam mengimplementasikan sensor tegangan digunakan rangkaian pembagi tegangan dan inputnya diberikan ke pin ADC dari arduino nano. Adapun perhitungan nilai ADC pada arduino sebagai berikut.

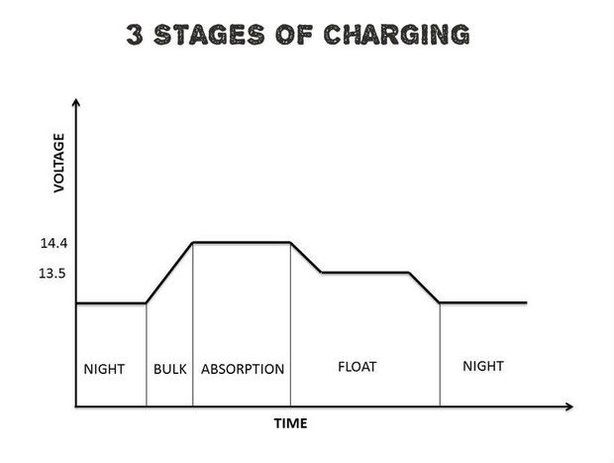
5V = ADC count 1024

1 ADC count = (5/1024)Volt= 0.0048828Volt

Vout=Vin\*R2/(R1+R2)

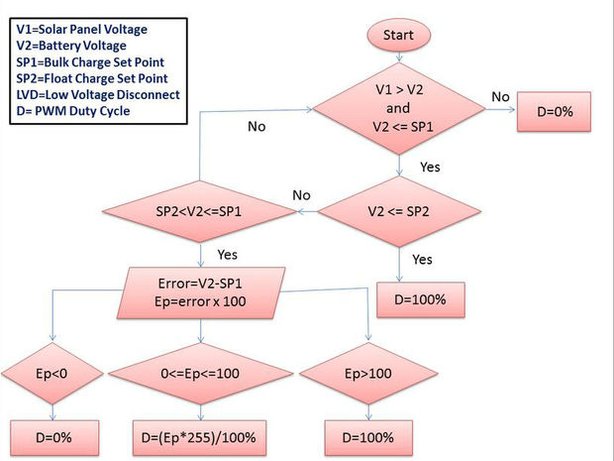
Vin = Vout\*(R1+R2)/R2 R1=100 and R2=20

Vin= ADC count\*0.00488\*(120/20) Volt



**Gambar Tahap Charging Baterai**

Pada *charge controller* ini, proses *charging* dibagi menjadi tiga tingkatan, yaitu *Bulk Charge, Absorption Charge,* dan *Float Charge.* Pada *bulk charge* arus dialirkan secara maksimum ke baterai sehingga tegangan baterai meningkat secara bertahap. Pada *Absorption Charge* tegangan baterai mencapai *bulk voltage.* Pada tahap tersebut, *charge controller* mempertahankan nilai PWM sehingga tegangan baterai konstan. Hal ini untuk mencegah baterai mengalami *over-heating.* Selanjutnya pada tahap *Float Charge,* baterai sudah dalam kondisi penuh sehingga diperlukan baterai untuk *discharge.* Dengan tahap *charging* seperti tersebut, dapat didesain software pada MCU untuk mengimplementasikan tahap *charging* tersebut*.* Flowchart software dapat dilihat pada gambar berikut.



**Gambar Flowchart Charge Controller**

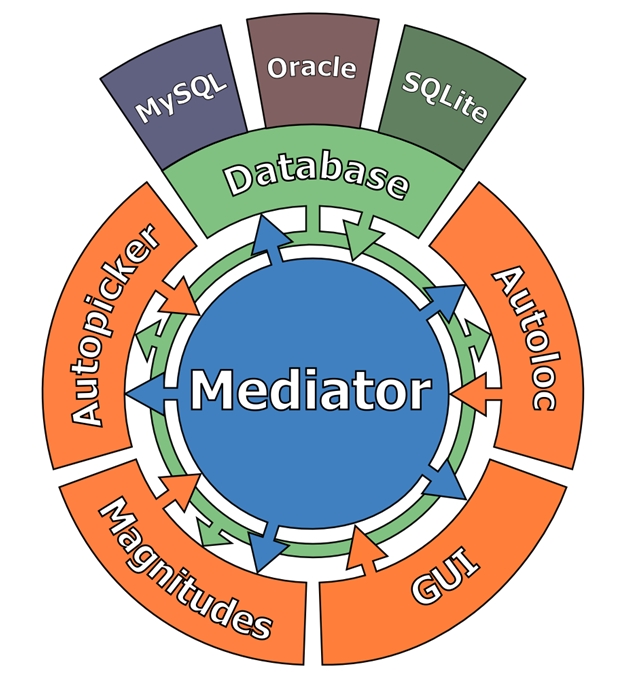
Hal pertama yang dilakukan adalah memeriksa apakah tegangan solar panel melebihi tegangan baterai dan tegangan baterai kurang dari tegangan *bulk-*nya. Jika tidak maka PWM akan memiliki duty cycle 0%. Jika iya, makadilakukan pemeriksaan lagi apakah tegangan solar panel kurang dari tegangan *float-*nya. Jika iya maka PWM akan menghasilkan sinyal dengan duty cycle sebesar 100%. Lalu tegangan baterai akan diperiksa apakah berada pada rentang tegangan *charging*-nya. Hasil perbedaan tegangan baterai dan tegangan *bulk* dijadikan nilai duty cycle dari nilai PWM yang dihasilkan.

## Desain Software

.

## Desain Pemodelan

A SeisComP3 automatic system consists of a set of independent applications each performing a discrete task. The communication between the applications is realized by a TCP/IP based messaging system. This messaging system is based on the open source toolkit “Spread” that provides a high performance messaging service across local and wide area networks. At the top of “Spread” a mediator, called scmaster handling additional requirements of SeisComP3 that are not natively provided by “Spread”. The messaging system is used for the exchange of meta data (e.g. picks) and administration of the program modules. The data model of SeisComP3 is based on the QuakeML schema version 0.5. QuakeML is also used as database object schema. By default SeisComP3 uses a MySQL database, but PostgreSQL is supported too.



The waveform data acquisition is based on the well established SeedLink protocol and the new ArcLink protocol both developed at the  GFZ Potsdam. The applications in SeisComP3 can be divided in four different groups: data acquisition, processing, graphical user interfaces and utilities. Detailed descriptions of the applications are in chapter 5. Table 4-1 shows a brief overview of the applications.

|  |  |  |
| --- | --- | --- |
| **Application** | **Type** | **Description** |
| Seedlink | data acquisition | providing realtime waveform data |
| Arclink | data acquisition | providing archive waveform data |
| scmaster | processing | handling messaging |
| scqc | processing | determination of waveform quality parameter |
| scautopick | processing | automatic picking |
| scautoloc | processing | automatic event detection and localization |
| scamp | processing | amplitude calculation |
| scmag | processing | magnitude calculation |
| scevent | processing | origin association, best, magnitude selection, best origin selection |
| scrttv | graphical user interface | real-time waveform monitor |
| scmv | graphical user interface | map overview showing,actual station status and events |
| scesv | graphical user interface | summary view of most important event information |
| scolv | graphical user interface | reviewing and revising origins, manual picking tool |
| scqcv | graphical user interface | showing station quality status |
| scmm | graphical user interface | message monitoring |
| scbulletin | utility | creating bulletins of events from the database |
| scdb | utility | inserting objects from QuakeML file or messaging into the database |
| scevtlog | utility | logging the event history |
| scevtls | utility | listing events for a given time range |
| scevtstreams | utility | listing all waveform streams used for event detection |
| scimex | utility | exchange of meta data objects between SeisComP3 systems with filter functionality |
| scimport | utility | forwarding of meta data objects from one messaging system to another |
| scm | utility | performance monitor similar to UNIX top |
| scproclat | utility | logging message history |
| scvoice | utility | event alert with optional voice output |
| scxmldump | utility | event dump to QuakeML file from database |
| sczip | utility | zip implementation of SeisComP3 |

### scevent

Associates Origins to Events or forms new Events if no suitable match is found. Selects preferred magnitude.

As a consequence of a real-time system the SeisComP3 system creates several origins (results of localization processes) for one earthquake because as time goes by more seismic phases are available. scevent receives these origins and associates the origins to events. It is also possible to import Origins from other agencies.

#### Origin Matching

Scevent associates Origins to Events by searching for the best match of the new (incoming) Origin to other Origins for existing Events. If a match is not found a new Event can be formed. The new Origin is matched to existing Origin by comparing location difference (horizontal only), Origin time difference, and matching Picks. The first best match is preferred where the options for an Origin match are (lowest to highest);

##### (1) Location and Time (lowest)

The difference in horizontal location is less than eventAssociation.maximumDistance (degrees) and the difference in Origin times is less than eventAssociation.maximumTimeSpan

##### (2) Picks

the two Origins have more than eventAssociation.minimumMatchingArrivals matching Picks. ***Note*** *(check this with Jan) it appears that Pick equality is based only on publicID. This effectively means that for Picks to be equal they will need to be created by the same picker configuration.*

##### (3) Picks and Location and Time (highest)

this is the best match, both of the Location-and-Time and Picks criteria above are satisfied.

Notes: for efficiency Events in the cache are scanned first and if no matches are found then the database is scanned for the time window eventAssociation.eventTimeBefore eventAssociation.eventTimeAfter around the incoming Origin time.

The order of objects in the cache will affect the first best match - is the time or insertion ordered?

***Possible Improvements*** *(Geoff and Rich to discuss with Jan) The location and time comparison could be improved by finding the Origin with the minium delta time and difference. The Pick comparison could be improved by fuzzy matching on Pick time, Name, and Channel.*

#### No Origin Match

If no Event with an Origin that matches the incoming Origin is found then a new Event is formed and the Origin is associated to that Event. The following criteria are applied to allow the creation of the new Event:

The Agency for the Origin is not black listed (processing.blacklist.agencies).

and

If the Origin is an Automatic then it has more than eventAssociation.minimumDefiningPhases Picks.

#### Preferred Origin

As already mentioned, one earthquake can be represented by several origins. So, one origin has to be chosen among the associated origins representing the location and time best. This is done by the following rules:

1. The latest manual or confirmed origin has the highest priority and is always preferred
2. Manual or confirmed origins have always a higher priority than automatic origins
3. In case the actual preferred origin is automatic, the incoming automatic origin will be preferred, if it contains more arrivals/phases and has a network magnitude being a possible preferred magnitude

#### Preferred Magnitude

The third task of scevent is to choose the preferred magnitude. For this purpose the following rules are used:

1. Network magnitudes are only valid if more than a defined number of station magnitudes exist (default=3). Mw(mB) is a strong motion magnitude resulting in special criteria: Mw(mB) is valid if equal or more than a defined amount of station magnitudes exist (default=8).
2. The priority of magnitudes is Mw(mB), MLv, mb (from high to low). For Mw(mB) additional priority criteria are defined.
   * If more than a defined number of station magnitudes exist for Mw(mB) (default = 30), Mw(mB) will always be preferred
   * If less than the defined number of station magnitudes exist for Mw(mB) (default = 30), Mw(mB) will be preferred when
     1. The sum of all stations magnitudes for Mw(mB) and for mb divided by 2 is more than a defined value (default = 6)
     2. If the number of station magnitudes for Mw(mB) is greater or equal than half of the station magnitude count for mb

#### Options

scevent supports configuration files (scevent.cfg).

### scautoloc

scautoloc is the SeisComP3 program responsible for automatically locating seismic events in near-real time. It normally runs as a daemon, continuously reading picks and amplitudes and processing them in real time. An offline mode is available as well. scautoloc reads automatic picks and several associated amplitudes. On that basis it tries to identify combinations of picks that correspond to a common seismic event. If the produced location meets certain consistency criteria, it is reported, i.e. passed on to other programs that take the origins as input.

#### Location procedure

The procedure of scautoloc to identify and locate seismic events basically consists of the following steps:

Pick preparation

In scautoloc each incoming pick needs to be accompanied by a specific set of amplitudes. Since in the SeisComP3 data model amplitudes and picks are independent objects, the amplitudes are added as attributes to their corresponding picks upon reception by scautoloc.

Pick filtering

Each incoming pick is filtered, i.e. it is checked if a pick is outdated and if the complete set of associated amplitudes is present already. If a station produces picks extremely often, these are considered to be more likely glitches and result in an increased SNR threshold.

Association

It is first attempted to associate an incoming pick with the known origins. Especially for large events with stable locations based on many picks already associated, this is the preferred way to handle the pick. If the association succeeds, the nucleation process can be bypassed. Under certain circumstances picks are both associated and fed into the nucleator.

Nucleation

If direct association fails, scautoloc tries to make a new origin out of this and other unassociated, previously received picks. This process is called “nucleation”. scautoloc performs a grid search over space and time, which is a rather expensive procedure as it requires lots of resources both in terms of CPU and RAM. Additional nucleation algorithms will become available in future. The grid is a discrete set of -in principle- arbitrary points that sample the area of interest sufficiently densely. In the grid search, each of the grid points is taken as a hypothetical hypocenter for all incoming picks. Each incoming pick is back projected in time for each of the grid points, on the assumption that it is a first-arrival “P” onset. If the pick indeed corresponds to a “P” arrival of a seismic event, and if this event was recorded at a sufficient number of stations, the back projected new pick will cluster with previous picks from the same event. The cluster will be densest around the origin time at the grid point closest to the hypocenter. In principle, the grid could be so dense that the location obtained from the grid search can be used directly. However, as RAM memory as well as CPU speed is limited, this is not possible. Therefore, if a cluster is identified as a potential origin, it does not necessarily mean that all contributing picks actually correspond to “P” arrivals. It may as well be a coincidental match caused by the coarseness of the grid or possible contamination by picked noise. Therefore, a location program (LocSAT) is run in order to try a location and test if the set of picks indeed forms a consistent hypocenter. If the pick residual RMS is too large, an improvement is attempted by excluding each of the contributing picks once to test if a reduction in RMS can be achieved. If the new origin meets all requirements, it is accepted as new seismic event location.  
The grid points are specified in a text file “grid.txt”. The default file shipped with scautoloc defines a grid with globally even distributed points at the surface, and depth points confined to regions of known deep seismicity. It may be modified, but should not comprise too many grid points (>3000, depending on CPU speed and RAM). See below for more details about the grid file.

Origin refinement

An origin produced or updated through association and/or nucleation may still be contaminated by phases wrongly interpreted as “P” arrivals. scautoloc tries to improve these origins based on e.g. pick SNR and amplitude. In this processing step, it is also attempted to associate phases which slipped through during the first association attempt, e.g. because the initial location was incorrect. If the origin contains a sufficient number of arrivals to assume a reasonably well location result, scautoloc additionally tries to associate picks as secondary phases such as “pP”. Such secondary phases are only “weakly associated”, i.e. these phases are not used for the location. For the analyst, however, it is useful to have possible “pP” phases predefined.

Origin filtering

This process involves final consistency checks of new/updated origins etc. During this procedure, the origins are not modified any more.

In the course of nucleation and association, as well as in the origin refinement and filtering, certain heuristic criteria are applied to compare the “qualities” of concurring origins. These criteria are combined in an internal origin score, which is based on properties of the picks themselves in the context of the respective origin (residuals, RMS, azimuthal gaps). In addition, the amplitudes provide valuable means of comparing origin qualities. Obviously, a pick with a high SNR will less likely be a transient burst of noise than a pick merely exceeding the SNR threshold. A high-SNR pick thus increases the origin score. Similarly, a pick associated to a large absolute amplitude is more likely to correspond to a real seismic onset, especially in case of simultaneous, large-amplitude observations at neighboring stations. A special case arises, when several nearby stations report amplitudes above a certain “XXL threshold”. For details see the section “Preliminary origins”. The amplitudes used by scautoloc are of type “snr” and “mb”, corresponding to the (relative, unit-less) SNR amplitude and the (absolute) “mb” amplitude, respectively. These two amplitudes are provided by [scautopick](https://www.seiscomp3.org/wiki/doc/applications/scautoloc#scautopick). In case of a setup in which scautopick is replaced by a different automatic picker, these two amplitudes must nevertheless be provided to scautoloc. Otherwise, the picks are not used. At the moment this is a strict requirement, in the future it may be changed.

The grid file

The grid configuration file consists of one line per grid point, each grid point specified by 6 columns, e.g.:

-10.00 105.00 20.0 5.0 180.0 8

The columns are grid point coordinates (latitude, longitude, depth), diameter, maximum station distance and minimum pick count, respectively. The above line sets a grid point centered at 10° S / 105° E at the depth of 20 km. It is sensitive to events within 5° of the center. Stations in a distance of up to 180° may be used to nucleate an event. At least 8 picks have to contribute to an origin at this location. The diameter should be chosen large enough to allow grid cells to overlap, but not too large. The size also determines the time windows for grouping the picks in the grid search. If the time windows are too long the risk of contamination with wrong picks increases. The maximum station distance allows to restrict to certain stations for the according grid points. E.g. stations from Australia are normally not required to create an event in Europe. If there is doubt, set the value to 180. The minimum pick count specifies how many picks are required for a given grid point to allow the creation of a new origin. The default grid file contains a global grid with even spacing of ~5° with additional points at greater depths where deep-focus events are known to occur.

The station configuration file

The station configuration file contains lines consisting of network code, station code, usage flag (0 or 1) and maximum nucleation distance. A usage flag of 1 indicates the station shall be used by scautoloc. If it shall not be used, 0 must be specified here. The maximum nucleation distance is the distance (in degrees) from the station up to which this station may contribute to a new origin. If this distance is 180°, this station may contribute to new origins world-wide. However, if the distance is only 10°, the range of this station is limited. This is a helpful setting in case of mediocre stations in a region where there are numerous good and reliable stations nearby. The station will then not pose a risk for locations generated outside the maximum nucleation distance. Network and station code may be wildcards (\*) for convenience. E.g.:

\* \* 1 90

GE \* 1 180

GE HLG 1 10

TE RGN 0 10

The example above means that all stations from all networks by default can create new events within 90°. The GE stations can create events at any distance, except for the rather noisy station HLG in the network GE, which is restricted to 10°. By setting the 3rd column to 0, TE RGN is ignored by scautoloc.

Preliminary origins

Usually, scautoloc will not report origins with less than a certain number of defining phases (specified by autoloc.minPhaseCount), typically 6-8 phases. In the case of events that result in very large amplitudes at a sufficient number of stations (hereafter called “XXL events”), it is possible to produce preliminary origins based on less picks.  
Prerequisite is that all these picks have extraordinary large amplitudes and SNR and lie within a relatively small region. Such picks are hereafter called “XXL picks”. A pick is internally tagged as “XXL pick” if its amplitude exceeds a certain threshold (specified by autoloc.thresholdXXL) and has a SNR > 8. For larger SNR picks with smaller amplitude can reach the XXL tag, because it is justified to treat a large-SNR pick as XXL pick even if its amplitude is somewhat below the XXL amplitude threshold. The XXL criterion should be judged as workaround to identify picks which justify the nucleation of preliminary origins.

Logging

scautoloc produces two kinds of log files: a normal application log file containing the processing and location history and an optional pick log. The pick log contains all received picks with associated amplitudes in a simple text file, one entry per line. This pick log should always be active as it allows pick playback for trouble shooting and optimization of scautoloc. If something did not work as expected, playing back the pick log will provide a useful way to find the source of the problem without the need of processing the raw waveforms again. The application log file contains miscellaneous information in variable format. The format of the entries may change anytime, so no downstream application should ever depend on it. There are some special lines, however. These contain certain keywords that allow convenient filtering of the most important information using grep. These keywords are NEW, UPD and OUT, for a new, updated and output origin, respectively. They can be used e.g. like

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

This will extract all lines containing the above keywords, providing a very simple (and primitive) origin history.

### scmag

Computes magnitudes.

The purpose of scmag is to compute magnitudes. It takes amplitudes and origins as input and produces StationMagnitudes and NetworkMagnitudes as output. The resulting magnitudes are sent to the “MAGNITUDE” group. scmag doesn’t access any waveforms. It only uses amplitudes previously calculated, e.g. by [scamp](https://www.seiscomp3.org/wiki/doc/applications/scamp). The purpose of scmag is the decoupling of magnitude computation from amplitude measurements. This allows several modules to generate amplitudes concurrently, like [scautopick](https://www.seiscomp3.org/wiki/doc/applications/scautopick) and [scamp](https://www.seiscomp3.org/wiki/doc/applications/scamp). As soon as an origin comes in, the amplitudes related to the picks are taken either from the memory buffer or the database to compute the magnitudes. Currently the following magnitude types are implemented:

MLv

Local magnitude calculated on the vertical component using a correction term to fit with the standard ML

MLh

Local magnitude calculated on the horizontal components to SED specifications. See [MLh](https://www.seiscomp3.org/wiki/doc/addons/MLsed)

mb

Narrow band body wave magnitude using a third order Butterworth filter with corner frequencies of 0.7 and 2.0 Hz.

mB

Broad band body wave magnitude.

Mwp

The body wave magnitude of Tsuboi et al. (1995)

Additionally, scmag computes the following derived magnitudes:

Mw(mB)

Estimation of the moment magnitude Mw based on mB using the Mw vs. mB regression of Bormann and Saul (2008)

Mw(Mwp)

Estimation of the moment magnitude Mw based on Mwp using the Mw vs. Mwp regression of Whitmore et al. (2002).

M(summary)

Summary magnitude, which consists of a weighted average of the individual magnitudes and attempts to be a best possible compromise between all magnitudes. See below for configuration and also [scevent](https://www.seiscomp3.org/wiki/doc/applications/scevent) for how to add the summary magnitude to the list of possible preferred magnitudes or how to make it always preferred.

Mw(avg)

Estimation of the moment magnitude Mw based on a weighted average of other magnitudes, currently MLv, mb and Mw(mB), in future possibly other magnitudes as well, especially those suitable for very large events. The purpose of Mw(avg) is to have, at any stage during the processing, a “best possible” estimation of the magnitude by combining all available magnitudes into a single, weighted average. Initially the average will consist of only MLv and/or mb measurements, but as soon as Mw(mB) measurements become available, these (and in future other large-event magnitudes) become progressively more weight in the average.

### scamp

scamp measures several different kinds of amplitudes from waveform data. It listens for origins and measures amplitudes in time windows determined from the origin. The resulting amplitude objects are sent to the “AMPLITUDE” messaging group. scamp is the counterpart of [scmag](https://www.seiscomp3.org/wiki/doc/applications/scmag). Usually, all amplitudes are computed at once by scamp and then published. Only very rarely an amplitude needs to be recomputed if the location of an origin changes significantly. The amplitude can be reused by scmag, making magnitude computation and update efficient. Currently, the automatic picker in SeisComP 3, [scautopick](https://www.seiscomp3.org/wiki/doc/applications/scautopick), also measures a small set of amplitudes (namely “snr” and “mb”, the signal-to-noise ratio and the amplitude used in mb magnitude computation, respectively) for each automatic pick in fixed time windows. If there already exists an amplitude, e.g. a previously determined one by scautopick, scamp will not measure it again for the respective stream. Amplitudes are also needed, however, for manual picks. scamp does this as well. Picks with weight smaller than 0.5 in the corresponding Origin are discarded.

Amplitudes for the following magnitudes are currently computed:

MLv

Local magnitude calculated on the vertical component using a correction term to fit with the standard ML.

MLsed

Local amplitude calculated on the horizontals. See [MLsed](https://www.seiscomp3.org/wiki/doc/addons/MLsed).

mb

Narrow band body wave magnitude using a third order Butterworth filter with corner frequencies of 0.7 and 2.0 Hz. Note that this amplitude is also computed by scautopick for all automatic picks.

mB

Broad band body wave magnitude.

Mw(mB)

Estimation of the moment magnitude Mw based on mB.

### scqc

scqc determines quality parameters of seismic data streams. The output parameters are time averaged quality control (QC) parameters in terms of waveform quality messages. In regular intervals report messages are sent containing the short term average representation of the specific QC parameter for a given time span. Alarm messages are generated if the short term average (e.g. 90s) of a QC parameter differs from the long term average (e.g. 3600s) more than a defined threshold.  
To avoid an excessive load, QC messages are sent distributed over time. The following parameters are determined:

Delay [s]

Time difference between arrival time and last record end time plus half record length (mean data latency, valid for all samples in record)

Latency [s]

Time difference between current time and record arrival time (feed latency)

Offset [counts]

Average value of all samples of a record

RMS [counts]

Offset corrected root mean square (RMS) value of a record

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

In case of the occurrence of a spike in a record this parameter delivers the time interval between adjacent spikes and the mean amplitude of the spike; internally a list of spikes is stored (spike time, spike amplitude); the spike finder algorithm is still preliminary

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

In case of a data gap between two consecutive records this parameter delivers the gap interval time and the mean length of the gap

Timing [%]

miniseed record timing quality (0 - 100 %)

#### Options

scqc supports command line options as well as configuration files (scqc.cfg).

#### Commandline

Command line options are limited to the default application options.

### scmaster

scmaster was designed as a kind of microkernel or mediator which delegates client requests. Therefore it is the key application responsible for the orchestration of the distributed system. In order to participate in the distributed system a client needs to send a connect request to the scmaster. In turn the master returns an acknowledgment message which either informs the client of its admission or rejection. If the connect request was successful the acknowledgment message will provide the client with the available message groups it can subscribe to. Moreover, all currently connected clients will be notified about the newly joined member. In case the master is configured with a database the client will also receive a direct follow up message which holds the address of this database. The address can be used to retrieve archived data later on. After a connection has been established every message will pass through the master first where it is processed accordingly and then relayed to the target groups. Once a client is done with processing a disconnect message will be sent to the master who in turn notifies all remaining clients about the leaving.  
scmaster can be configured with a database to ensure the integrity of the system. Before a message is distributed by scmaster the message is written to the specified database. This way each message is stored before it enters the system. In case of a crash all necessary information can be recovered from the database. Currently, driver exist for MySQL, PostgreSQL and sqlite. Note that the scmaster can run without a database but loses data integrity in doing so.

#### Options

scmaster supports commandline options as well as configuration files (scmaster.cfg).

#### Configuration

Users configuration file: $HOME/.seiscomp3/scmaster.cfg

msgGroups = <list>

message groups listed will be created. By default the following groups will be created:

* STATUS\_GROUP
* IMPORT\_GROUP
* AMPLITUDE
* CONFIG
* EVENT
* GUI
* INVENTORY
* LOCATION
* LOGGING
* MAGNITUDE
* PICK
* PUBLICATION
* QC
* SERVICE\_PROVIDE
* SERVICE\_REQUEST

### SeedLink

SeedLink? is a real-time data acquisition protocol and a client-server software that implements this protocol. The SeedLink? protocol is based on TCP. All connections are initiated by the client. During handshaking phase the client can subscribe to specific stations and streams using simple commands in ASCII coding. When handshaking is completed, a stream of SeedLink? “packets” consisting of a 8-byte SeedLink? header (containing the sequence number) followed by a 512-byte Mini-SEED record, is sent to the client. The packets of each individual station are always transferred in timely (FIFO) order. The SeedLink? implementation used in SeisComP is the oldest and most widely used, however, other implementations exist. Another well-known implementation is deployed in IRIS DMC and some manufacturers have implemented SeedLink? in their digitizer firmware. All implementations are generally compatible, but not all of them support the full SeedLink? protocol. On the other hand IRIS DMC implements some extensions which are not supported by other servers. In the following we use ”SeedLink?” to denote the SeedLink? implementation used in SeisComP. The data source of a SeedLink? server can be anything which is supported by a SeedLink? plug-in - a small program that sends data to the SeedLink? server. Plug-ins are controlled by the SeedLink? server, e.g., a plug-in is automatically restarted if it crashes or a timeout occurs. Data supplied by a plug-in can be a form of Mini-SEED packets or just raw integer samples with accompanying timing information. In the latter case, the SeedLink? server uses an integrated “Stream Processor” to create the desired data streams and assemble Mini-SEED packets.

|  |  |
| --- | --- |
| **Digitizer/DAS** | **Plugin Implementer** |
| SeedLink? | GFZ |
| LISS | Chad Trabant (IRIS) |
| Quanterra Q330 | Jet Spring, Inc.; ISTI, Inc.; Chad Trabant (IRIS); GFZ |
| Quanterra Q380/Q680, Q4120, Q720 (not supported by SeisComp? 3.0) | GFZ (based on Comserv by Quanterra, Inc.) |
| Earth Data PS2400/PS6-24 | GFZ |
| Lennartz M24 | Lennartz Electronic GmbH |
| Geotech DR24 | GFZ |
| Nanometrics HRD24 | GFZ; Recai Yalgin |
| Guralp DM24 | GFZ (based on libgcf2 from Guralp) |
| SARA SADC10/18/20/30 | GFZ |
| RefTek? RTPD | GFZ (based on software library provided by RefTek?, Inc.) |
| NRTS | GFZ (based on ISI toolkit from David E. Chavez) |
| NAQS | Chad Trabant (IRIS; based on sample code from Nanometrics, Inc.); Matteo Quintiliani (INGV; nmxptool) |
| SCREAM | Reinoud Sleeman (KNMI) |
| Earthworm | Chad Trabant (IRIS) |
| Antelope | Chad Trabant (IRIS) |
| WIN | GFZ (based on source code of WIN system) |
| Lacrosse 2300 Weather Station | GFZ (based on open2300 library from Kenneth Lavrsen) |
| Reinhardt MWS5/MWS9 Weather Station | GFZ |
| Generic MODBUS/TCP devices | GFZ |

**Supported data sources**  
  
Table 5-1 lists digitizers and data acquisition systems that are supported by SeedLink? plug-ins. More plug-ins (Kinemetrics K2, Lennartz MARS-88, Lennartz PCM 5800, etc.) have been implemented by various users, but are not (yet) included in the package. The included C language plug-in interface is described in section 5.1.1.5. Antelope, Earthworm and NAQS can also import data from SeisComP. In SeisComP the class RecordStream is implemented that supports both SeedLink? and [ArcLink](https://www.seiscomp3.org/wiki/doc/applications/seedlink#ArcLink) sources; this class is used by all SeisComP modules that work with waveform data. On a lower level, SeedLink? clients can be implemented using the libslink software library or its Java counterpart, JSeedLink. Libslink supports Linux/UNIX, Windows and !MacOS X platforms, and comes with an exhaustive documentation in form of UNIX manual pages.

#### Protocol description

A SeedLink? session starts with opening the TCP/IP connection and ends with closing the TCP/IP connection. During the session the following steps are performed in order:

* opening the connection
* handshaking
* transferring SeedLink? packets

We will take a closer look at the protocol. Note, the details are normally hidden from the clients by the libslink software library; therefore it is not necessary to be familiar with the protocol in order to implement clients.

#### Handshaking

When the TCP/IP connection has been established the server will wait for the client to start handshaking without initially sending any data to the client. During handshaking the client sends SeedLink? commands to the server. The commands are used to set the connection into a particular mode, setup stream selectors, request a packet sequence number to start with and eventually start data transmission. SeedLink? commands consist of an ASCII string followed by zero or several arguments separated by spaces and terminated with carriage return (<cr>, ASCII code 13) followed by an optional linefeed (<lf>, ASCII code 10). The commands can be divided into two categories: “action commands” and “modifier commands”. Action commands perform a function such as starting data transfer. Modifier commands are used to specialize or modify the function performed by the action commands that follow. When a server receives a modifier command it responds with the ASCII string “OK” followed by a carriage return and a line feed to acknowledge that the command has been accepted. If the command was not recognized by the server or has invalid parameters, then the ASCII string “ERROR” is sent as a response to the client followed by a carriage return and a line feed. The client should not send any further commands before it has received a response to the previous modifier command. If a network error or timeout occurs the client should close the connection and start a new session..

## Desain Basis Data