

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Menurut Asbon Hendra (2012 : 157) Sistem merupakan kumpulan dari user atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Sedangkan,

Menurut Indrajani (2011: 48) pengertian sistem yaitu Sekelompok komponen yang saling berhubungan dan berkerjasama untuk mencapai satu tujuan yang sama dengan menerima *input* dan menghasilkan *output* dalam proses transformasi yang teratur.

Kesimpulan dari sistem diatas adalah sebagai berikut: Suatu pola korelasi antar prosedur yang saling terkait satu sama lain .pada hubungan tersebut merupakan poin penting terhadap bekerjanya elemen-elemen untuk menyelesaikan suatu sasaran. Bagian-bagian atau elemen-elemen yang saling terintegrasi satu dengan yang lainnya untuk mencapai suatu tujuan yang sama.

II.1.2. Karakteristik Sistem

Menurut Asbon Hendra (2012 : 158) karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem atau bagian-bagian dari sistem. Setiap sistem, tidak peduli betapa pun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut “supra sistem”.

2. Batasan Sistem (Boundary)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan yang lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

3. Lingkungan Luar Sistem (Environment)

Environment merupakan segala sesuatu di luar batas sistem yang memengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan merugikan.

4. Penghubung Sistem (Interface)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem lainnya. Dengan kata lain, output dari suatu subsistem akan menjadi input dari subsistem lainnya.

5. Masukan Sistem (Input)

Merupakan energi yang di masukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (Signal input) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (Output)

Merupakan hasil dari energi yang di olah oleh sistem, meliputi output yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Dan output yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

7. Pengolah Sistem (Process)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang di inginkan. Contoh CPU pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi, serta bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Tujuan Sistem (Goal)

Tujuan sistem adalah suatu sistem pasti memiliki suatu tujuan ataupun sasaran yang memengaruhi input yang di butuhkan dan output yang

dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran dan tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya.

Kesimpulan adanya tujuan sistem adalah sebagai berikut: Merupakan dasar utama atas diciptakan suatu sistem. Tanpa tujuan ataupun sasaran suatu sistem tidak akan dapat diukur berhasil atau tidaknya jika dioperasikan. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak berguna sama sekali. dengan kata lain sistem dikatakan berhasil kalo pengoperasian mengenai sasaran.

II.2. Pengertian Informasi

Menurut Asbon Hendra (2012 : 167) Informasi adalah merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi = input – proses – output. Sedangkan,

Menurut Kusrini (2010:7), Informasi adalah data yang diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar bila dibandingkan biaya untuk mendapatkannya.

Kesimpulan Informasi adalah sebagai berikut: Informasi dapat dikatakan sebagai hasil dari pengolahan data. dalam proses perubahan tersebut diharapkan bisa menjadi fakta dan nilai yang memiliki arti juga bermanfaat. Data

yang diproses menjadi bentuk yang sangat berarti bagi penerima karena berupa fakta dan nilai yang memiliki arti bermanfaat.

II.3. Sistem Informasi

Menurut Eddy Prahasta (2014 : 70-78) Sistem Informasi dapat didefinisikan sebagai sekumpulan objek, ide dan data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya, berikut keterkaitannya di dalam mencapai tujuan. Dengan kata lain, sistem informasi adalah sekumpulan komponen (sub-sistem fisik & non-fisik) dan informasi yang saling berhubungan satu sama lainnya dan bekerja sama untuk mencapai suatu tujuan. Sedangkan,

Menurut Kusrini (2010:8), Sistem informasi adalah suatu system dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dan laporan-laporan yang diperlukan.

Kesimpulan Sistem informasi adalah sebagai berikut: Sekumpulan subsistem dan informasi yang memiliki korelasi antar bagian objek, ide, dan data merupakan bentuk konkrit yang diperoleh penerima sebagai bagian dari proses pencapaian tujuan tertentu. Sistem informasi merupakan sekumpulan sebuah objek, data, dan ide untuk yang ditempatkan pada penerima yang saling berhubungan untuk mencapai suatu tujuan.

II.3.1. Kualitas Informasi

Menurut Kusrini (2010:8), Informasi yang berkualitas memiliki 3 kriteria yaitu :

a. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

b. Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat. Didalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal ini dapat berakibat batal pada perusahaan.

c. Relevan (*Relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. Disamping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (*Value of Information*) ditentukan oleh 2 hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaat lebih besar disamping biaya untuk mendapatkannya.

II.4 Sistem Informasi Geografis

Menurut Edi Irwansyah (2010:1), Sistem Informasi Geografis atau *Geografic Information System* adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, mengatur dan menampilkan seluruh jenis geografis. Sedangkan

Menurut Eddy Prahasta (2014 : 95) Sistem Informasi Geografis merupakan sejenis perangkat lunak, perangkat keras, manusia, prosedur, basis data, dan fasilitas jaringan komunikasi yang digunakan untuk memfasilitasi proses-proses pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran data/informasi geografis.

Kesimpulan sistem informasi geografis adalah sebagai berikut: Sekumpulan jenis perangkat lunak, perangkat keras, manusia, prosedur, data dan fasilitas jaringan dan mempunyai proses pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran data beserta atributnya. Sistem informasi geografis yang sudah terintegrasi yang dalam pengembangan suatu model dan merupakan sistem yang baik sekali.

II.4.1 Komponen Sistem Informasi Geografis

Menurut Eddy Prahasta (2014 : 104) Sistem Informasi Geografis merupakan sistem kompleks yang umumnya terintegrasi dengan sistem komputer lainnya di tingkat fungsional & jaringan. Data raster yang digunakan yaitu peta kabupaten karo. Data atributnya yaitu kode, nama desa,

kecamatan, nama sensor, jenis sensor dan. Dan aplikasi yang digunakan Mapinfo dan servernya adalah localhost.

a) Perangkat keras (*Hardware*)

SIG tersedia di berbagai *platform* perangkat keras; mulai dari kelas PC *desktop*, *workstations*, hingga *multi-user host*. Walaupun demikian, fungsionalitas SIG tidak terikat ketat pada karakteristik fisik perangkat kerasnya hingga keterbatasan memori pada PC dapat diatasi. Adapun perangkat keras yang sering digunakan untuk aplikasi SIG adalah komputer (PC/CPU), *mouse*, *keyboard*, monitor (plus VGA-card grafik) yang beresolusi tinggi, *digitizer*, *printer*, *plotter*, *receiver* GPS dan *scanner*.

b) Perangkat lunak (*Software*)

SIG merupakan sistem perangkat lunak dimana sistem basisdatanya memegang peranan kunci. Pada SIG lama, sub-sistem diimplementasikan oleh modul-modul perangkat lunak hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program (*.exe) yang dapat dieksekusi tersendiri.

c) Data & informasi geografis

SIG dapat mengumpulkan & menyimpan data/informasi yang diperlukan baik tidak langsung (dengan meng-*import*-nya) maupun langsung dengan mendijitasi data spasialnya (*on-screen/head-ups* pada layar monitor atau cara manual dengan *digitizer*) dari peta analog dan memasukkan data atributnya dari tabel/laporan dengan menggunakan *keyboard*.

d) Manajemen

Proyek SIG akan berhasil jika dikelola dengan baik & dikerjakan oleh orang yang memiliki keahlian yang tepat pada semua tingkatan. Data raster yang digunakan yaitu peta kabupaten karo. Data atributnya yaitu kode, nama desa, kecamatan, nama sensor, jenis sensor dan. Dan aplikasi yang digunakan *Map info* dan servernya adalah localhost.

II.5. PHP

Menurut Loka Dwiartara (2012 : 1) PHP (PHP: Hypertext Preprocessor) adalah bahasa script yang ditanam di sisi server. Prosesor PHP dijalankan di server (*Windows atau Linux*). Saat sebuah halaman dibuka dan mengandung kode PHP, prosesor itu akan menerjemahkan dan mengeksekusi semua perintah dalam halaman tersebut, dan kemudian menampilkan hasilnya ke browser sebagai halaman HTML biasa. Karena penerjemahan ini terjadi di server, sebuah halaman ditulis dengan PHP dapat dilihat dengan menggunakan semua jenis browser, di sistem operasi apaun. Sedangkan

Menurut Andrea Edelheid dan Khairil Nasution (2012:1), PHP atau *Hipertext Processor* merupakan bahasa berbentuk *script* yang ditempatkan dalam server dan dieksekusi dalam server dan selanjutnya ditransfer dan dibaca oleh client. PHP juga disisipkan dalam Bahasa HTML.

Kesimpulan PHP adalah sebagai berikut: PHP adalah bahasa pemrograman berbasis web dan mendukung dan baik sekali dalam penanganan database. Bahasa induk PHP adalah CC++ Menjadikan PHP untuk mudah

dipelajari oleh berbagai kalangan. Dan dengan PHP halama web menjadi lebih Dinamis dan Interaktif.

II.5.1. Sejarah Singkat PHP

PHP pertama kali ditemukan pada 1995 oleh seorang Software Developer bernama Rasmus Lerdrof. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. Script yang dikembangkan baru dapat melakukan 2 pekerjaan, yakni merekam informasi visitor, dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang dimilisi mendiskusikan script buatan Rasmus Lerdrof, hingga akhirnya Rasmus mulai membuat sebuah tool/script, bernama Personal Home Page (PHP) (Loka Dwiartara : 3).

II.5.2. Keunggulan-Keunggulan PHP

Menurut Adhi Prasetyo (2012 : 4-5) Pada saat ini banyak website yang menggunakan PHP sebagai dasar pengolahan data. Beberapa keunggulan yang dimiliki PHP adalah:

1. Kesederhanaan, karena PHP memang secara khusus didesain untuk membuat aplikasi web, dan karena itu, PHP memiliki banyak sekali fungsi *built in* untuk menangani kebutuhan standar pembuatan aplikasi web. Dengan adanya fungsi-fungsi tersebut, maka tentu saja proses belajar PHP terutama dalam pengembangan aplikasi akan jauh lebih mudah karena semua sudah tersedia.

2. PHP bersifat *open source*. Karena *source code* PHP tersedia secara gratis, maka hal tersebut memungkinkan komunitas *developer* untuk selalu melakukan perbaikan, pengembangan dan menemukan *bug* dalam bahasa PHP.
3. Stabilitas dan kompatibilitas. Saat ini, PHP berjalan stabil di berbagai macam sistem operasi seperti berbagai versi UNIX (termasuk *Linux*), *Windows* dan *Mac*. PHP juga terintegrasi secara baik dengan berbagai macam web server termasuk dua yang paling populer yaitu *IIS* dan *Apache*.
4. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti *support* langsung ke berbagai macam *database* yang populer, arsitektur yang dapat dikembangkan dan sebuah prosesor yang tidak hanya menggunakan *resource* minimal pada komputer anda dibandingkan kompetitornya, tetapi juga dapat menampilkan halaman web dengan cepat.

Kesimpulan keunggulan PHP adalah sebagai berikut: Keunggulan php secara garis yakni sederhana, bersifat open source, stabilitas dan kompatibilitas , dan dilengkapi berbagai macam pendukung. PHP sangat fleksibel dan akan memberikan kemudahan dalam aplikasinya. Misalnya data yang dimasukkan kedalam form html secara otomatis divariabelkan dan dapat digunakan langsung sehingga tidak perlu memparsing apa yang disebut *Query String*.

II.6. HTML

Menurut Adhi Prasetyo (2015 : 98) HTML (*Hyper Text Markup Language*) merupakan sebuah file teks yang berisi *tag-tag markup* yang dapat memberitahukan browser bagaimana harus menampilkan sebuah halaman. *File* HTML harus memiliki ekstensi htm atau html dan *file* HTML juga dapat dibuat menggunakan editor teks seperti *Notepad* ataupun *Notepad++*.

Kesimpulan HTML adalah sebagai berikut: HTML digunakan untuk membuat halaman website atau sebuah website. HTML juga disebut dengan bahasa Markup. Dalam pembuatan halaman web HTML bisa menggunakan notepad untuk menuliskan perintah kekomputer.seingga komputer nantinya akan menafsirkan perintah tersebut ke bentuk halaman web.

II.7. Mapinfo

Map Info adalah satu software pengolah Sistem Informasi Geografi (SIG). MapInfo diminati oleh pemakai SIG karena mempunyai karakteristik yang menarik, seperti mudah digunakan, harga relatif murah, tampilan interaktif, *user friendly*, dan dapat dimodifikasi menggunakan bahasa *script* yang dimiliki. Kemudahan lain MapInfo tidak memerlukan dukungan *hardware* yang terlalu tinggi, sehingga hampir semua spesifikasi komputer dapat menggunakan MapInfo. *Software* ini memiliki kemampuan untuk mengorganisir, memanipulasi, serta menganalisa data (Elex Media Komputindo, 2015 : 1).

II.8. MySQL

Menurut Adhi Prasetyo (2015 : 199) MySQL adalah sebuah server *database open source* yang memiliki struktur relasional, ada tabel-tabel yang menyimpan data, Setiap tabel terdiri dari kolom dan baris. Sedangkan

Menurut Wahana Komputer (2010:5), Mysql adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. Mysql memiliki dua bentuk lisensi yaitu *free software* dan *shareware*.

Kesimpulan MySQL adalah sebagai berikut: SQL adalah bahasa query yang standard yang digunakan sebagai suatu bahasa sederhana dan dasar, yang memungkinkan anda untuk berkomunikasi dengan database, membaca, menulis, dan memperoleh informasi yang berguna database. RDBMS Merupakan terobosan dari database management system yang mengorganisasikan data dalam suatu struktur dan memaksimalkan berbagai cara serta menghubungkan antar kumpulan data yang disimpan dalam database.

II.9. WWW (*Word Wide Web*)

Menurut Asbon Hendra (2012 : 145) *Web* adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet webserver dipresentasikan dalam bentuk *hypertext*. Informasi di *web* dalam bentuk teks umumnya ditulis dalam format HTML (*Hypertext Markup Language*). Sedangkan

Menurut Janner Simarmata (2010:59), Aplikasi Web adalah aplikasi yang sejak awal dirancang untuk dieksekusi didalam program yang berbasis web. Defenisi ini mengungkapkan 2 aspek penting dari aplikasi yaitu :

- a. Suatu aplikasi web dirancang agar dapat berjalan dilingkungan berbasis web. Artinya aspek-aspek hipermedia dalam kaitanya dengan *hypertext* dan multimedia didalan kombinasi dengan logika aplikasi tradisional harus diperhitungkan di seluruh siklus hidup aplikasi, yang membuatnya berbeda dengan aplikasi konvensional.
- b. Aplikasi web adalah suatu aplikasi yang tidak berupa sekumpulan halaman-halaman web.
- c. Secara khusus aplikasi web menguatkan notasi sesi membedakannya dari paradigma web permintaan respons (*request reponse*) yang biasa. Dalam konteks ini, web services secara dinamis akan menghasilkan halaman yang tidak mungkin dipertimbangkan aplikasi web. Contoh, suatu layanan yang jadwal yang diberi tujuan dan keberangkatan, dan tempat yang diinginkan akan mengembalikan sekumpulan halaman yang berisi koneksi dan kereta yang tersedia.

Kesimpulan WWW (*Word Wide Web*) adalah sebagai berikut: *Web* adalah sebuah penyebara informasi dari melalui internet. Sistem informasi yang berbentuk dalam teks, gambar, suara dan lain-lain dan tersimpan sebuah internet *webserver* dan dipresentasikan dalam bentuk *hypertext*.

II.10. Gempa Bumi

Menurut Primus Supriyono (2014 : 3) Gempa Bumi adalah peristiwa bergetarnya akibat pelepasan energi dari dalam perut bumi secara tiba-tiba sehingga menciptakan gelombang seismik yang ditandai dengan patahnya lapisan batuan kerak bumi. Sedangkan

Menurut Muhammad Andang Novianta (Simposium Nasional RAPI XI FT UMS-2012) Gempaan Bumi adalah gempa bumi yang terjadi bergesernya lempeng pada permukaan bumi.

Kesimpulan gempa bumi adalah sebagai berikut: Gempa bumi, getaran yang terjadi permukaan bumi dan biasa disebabkan oleh pergerakan lempeng bumi. Gempa bumi disebabkan oleh pelepasan energi yang dihasilkan oleh tekanan yang dilakukan lempengan yang bergerak, semakin lama maka tekanan itu akan semakin membesar dan akhirnya mencapai pada keadaan dimana tekanan tersebut tidak dapat ditahan oleh pinggiran lempengan, maka terjadilah gempa bumi.

II.10.1. Proses Gempa Bumi

Gempa Bumi sebenarnya terjadi tidak secara tiba-tiba, tetapi melalui proses atau tahap-tahap tertentu. Proses terjadinya gempa bumi tektonik dan vulkanik.

a) Gempa Bumi Tektonik

Gempa bumi tektonik terjadi karena gesekan atau tumbukan pada kerak bumi. Tumbukan ini terjadi karena adanya pergerakan antar lempeng-lempeng tektonik yang berada dibawah permukaan bumi.

b) Gempa Bumi Vulkanik

Gempa bumi vulkanik terjadi karena adanya pergerakan secara terus-menerus dari magma dan cairan yang bersifat hidrotermal (peka terhadap panas) dibawah gunung berapi.

II.10.2. Penyebab Gempa Bumi

Menurut Primus Supriyono (2014 : 12-15) Ada beberapa faktor yang menyebabkan terjadinya gempa bumi. Penyebab-penyebab terjadinya gempa bumi antara lain karena pelepasan energi lempeng-lempeng tektonik, proses subdiksi, pergerakan magma, penumpukan massa air. Injeksi atau akstraksi cairan, atau karena pengguna bahan peledak.

a) Pelepasan Energi Lempeng Tektonik

Pelepasan energi tektonik karena adanya tekanan yang dilakukan oleh pergerakan lempeng-lempeng tektonik secara terus-menerus. Semakin lama tekanan itu akan semakin besar, yang akhirnya tekanan tersebut tidak mampu ditahan lagi oleh pinggiran lempeng-lempeng bumi. Pada saat itulah terjadi pelepasan energi secara tiba-tiba sehingga mengakibatkan gempa bumi.

b) Proses Subduksi

Dalam proses ini, terjadi dua tumbukan antara dua lempeng bumi, dimana salah satu lempeng bumi terdorong kebawah lempeng bumi yang lain.

c) Pergerakan Magma

Jenis gempa bumi yang lain juga dapat terjadi karena pergerakan magma didalam gunung berapi. Gempa bumi seperti itu dapat menjadi air

d) Penumpukan Massa Air

Jenis gempa bumi yang lain dapat terjadi karena menumpuknya massa air yang sangat besar dibalik dam.

e) Injeksi atau Akstraksi Cairan

Sebagian lagi gempa bumi juga dapat terjadi karena injeksi atau akstraksi cairan dari atau kedalam bumi.

f) Penggunaan Bahan Peledak

Jenis gempa bumi yang lain dapat terjadi karena aktivitas peledakan menggunakan bahan peledak dengan kekuatan yang besar.gempa bumi disebabkan oleh aktifitas manusia seperti ini dinamakan seismisitas terinduksi.penggunaan bahan peledak pada aktifitas industri pertambangan dapat menyebabkan terjadinya gempa bumi.

II.11. Sensor

Sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu energi. seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya.

II.11.1. Sensor Getar (*vibration sensor*)

Menurut (Muhammad Andang Novianta, Simposium Nasional RAPI XI FT UMS-2012) Sensor getar adalah merupakan sensor yang berfungsi untuk mendeteksi sinyal perambatan getaran gempa dalam arah gelombang horizontal maupun vertikal dan mengkonversi sinyal yang getaran yang terdeteksi menjadi besaran listrik analog.

Kesimpulan Sensor Getar adalah sebagai berikut: Proses pendeteksian sinyal getaran dalam arah vertikal maupun arah horizontal. Dan Validasi sinyal keluaran sensor getaran dapat diatur secara tepat selama Vaktor penguatan dari setiap penguat yang digunakan didalam sistem memiliki Faktor maksimal sebesar 1x. Sistem pendeteksi gempa bumi mikrokontroler terbukti dapat digunakan untuk mendeteksi taraf getaran yang sangat kecil dan menvesualisasikan sinyal gelombang.

II.11.2. Seismograf

Menurut Primus Supriyono (2014 :20) Seismograf adalah meruapakan alat yang dapat mengukur kekuatan gempa dan dapat mencatat lama dan intensitas

gelombang atau getaran gempa. Seismograf Modern dapat menangkap dan mencatat gelombang baik dari arah horizontal maupun vertikal.

II.12. UML (*Unified Modelling Language*)

Menurut Adi Nugroho (2010 : 6-7) UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Sedangkan

Menurut Julius Hermawan (2010:13), UML menyediakan beberapa notasi dan artifak standard yang bisa digunakan sebagai alat komunikasi bagi para proses analisis dan desain.

Kesimpulan UML adalah sebagai berikut: Sistem atau perangkat lunak yang berorientasi objek yang sesungguhnya digunakan untuk penyederhanaan permasalahan yang kompleks sehingga lebih mudah dipelajari dan dipahami. Sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan membangun dan mendokumentasikan sebuah sistem pengembangan perangkat lunak berbasis objek.

Tabel II.1. View dan Diagram dalam UML

Major Area	View	Diagrams	Main Concepts
Structural	Static view	Class diagram	Class, association, generalization, dependency, realization, interface.
	Use case view	Use case diagram	Use case, actor, association, extend, include, use case generalization
	Implementation view	Component diagram	Component, interface, dependency, realization
	Deployment view	Deployment diagram	Node, component, dependency, location
Dynamic	State machine	Statechart diagram	State, event, transition, action
	Activity view	Activity diagram	State, activity, completion transition, fork, join
	Interaction view	Sequence diagram	Interaction, object, message, activation
		Collaboration diagram	Collaboration, interaction, collaboration role, message
Model management	Model management view	Class diagram	Package, subsystem, model
Extensibility	All	All	Constraint, stereotype, tagged values

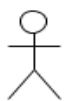
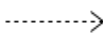



(Sumber : Adi Nugroho ; 2010)

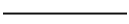
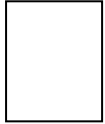



II.12.1. Use Case Diagram

Menurut Adi Nugroho (2010 : 6-7) *Use Case Diagram* merupakan unit fungsionalitas koheren yang diekspresikan sebagai transaksi yang terjadi antara actor dan sistem. *Use Case* sesungguhnya merupakan unit koheren dari fungsionalitas sistem/perangkat lunak yang tampak dari luar dan diekspresikan sebagai urutan pesan – pesan yang dipertukarkan unit – unit sistem dengan satu atau lebih actor yang ada diluar sistem. Sedangkan

Menurut Julius Hermawan (2010:13) Use case menjelaskan urutan kegiatan yang dilakukan Aktor dan system untuk mencapai tujuan tertentu.

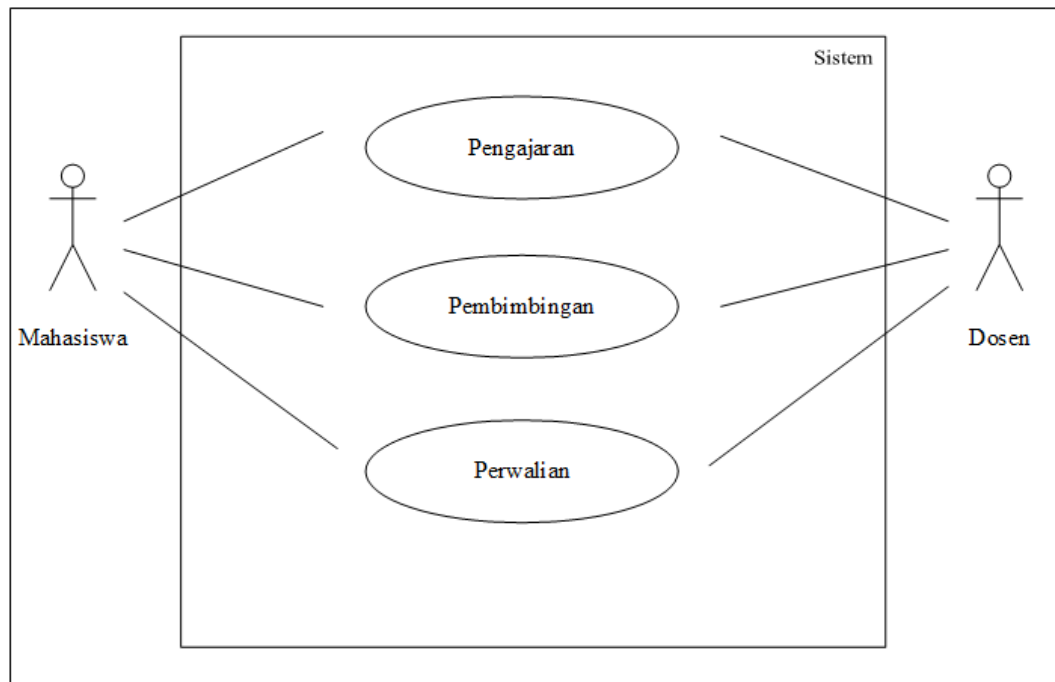
Tabel II.2 Komponen Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Grady Booch ; 2013)

Kegunaan dari *use case* adalah untuk mendaftarkan *actor-actor* dan *use case-use case* dan memperlihatkan *actor-actor* mana yang berpartisipasi dalam masing-masing *use case*.



Gambar II.1 Use Case

(Sumber : Adi Nugroho ; 2010 : 34)

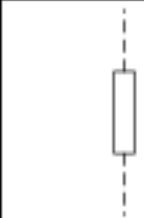
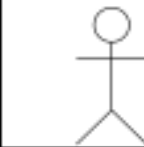
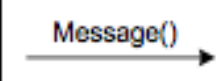



Kesimpulan *Use Case Diagram* adalah Sebagai berikut: Unit fungsional antara aktor dan sistem yang saling terkait dan konsisten yang terjadi diluar sistem tersebut. Dan Use Case Diagram dapat menggambarkan hubungan use case dengan actor.

II.12.2. *Sequence Diagram*

Menurut Adi Nugroho (2010 : 6-7) *Sequence Diagram* memperlihatkan interaksi sebagai diagram dua matra (dimensi). Matra vertikal adalah sumbu waktu bertambah dari atas ke bawah. Matra horizontal memperhatikan peran pengklasifikasi yang merepresentasikan objek-objek mandiri yang terlibat dalam kolaborasi. Masing-masing peran pengklasifikasikan direpresentasikan sebagai

kolom–kolom vertikal dalam *sequence diagram* sering disebut garis waktu (*lifeline*). Selama objek ada, peran digambarkan menggunakan garis tegas. Selama aktivasi prosedur pada objek aktif, garis waktu digambarkan sebagai garis ganda. Pesan-pesan digambarkan sebagai suatu tanda panah dari garis waktu suatu objek ke garis waktu objek lainnya. Panah-panah yang menggambarkan aliran pesan antarperan pengklasifikasi digambarkan dalam urutan waktu kejadiannya dari atas ke bawah.

Tabel II.3 Komponen Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
		<i>Actor</i>	Digunakan untuk menggambarkan user / pengguna.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form.
4		<i>Control Class</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
5		<i>Entity Clas</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

(Sumber : Grady Booch ; 2013)



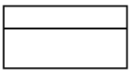

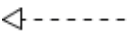
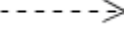

Kesimpulan *sequence diagram* adalah sebagai berikut: Menggambarkan interaksi antar objek didalam dan sekitar sistem pengguna, display, dan *sequence diagram* terdiri dari horizontal dan vertikal . *Sequence diagram* biasa digunakan untuk skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu.

II.12.3. *Class Diagram*

Menurut Adi Nugroho (2010 : 6-7) *Class Diagram* adalah kelas yang sesungguhnya mempresentasikan suatu kodi dalam konsep diskret di dalam aplikasi yang dimodelkan, sesuatu yang bersifat fisika (misalnya mobil, pesawat terbang dan sebagainya), sesuatu yang bersifat bisnis (misalnya pesanan), sesuatu yang bersifat logika (misalnya penjadwalan), sesuatu yang sangat berkait dengan aplikasi (misalnya tombol-tombol, ikon-ikon, dan sebagainya), sesuatu yang merupakan konsep yang dikenali dalam terminalogi ilmu komputer (misalnya tabel hash atau berbagai metode pengurutan dan pencarian (sorthing dan serching), atau sesuatu yang bersifat perilaku (behaviour) misalnya pekerjaan tertentu. Sedangkan

Menurut Julius Hermawan (2010:13) *Class Diagram* merupakan pembentuk utama dari system berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama.

Tabel II.4 Komponen Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Grady Booch ; 2013)






Kesimpulan *class diagram* adalah sebagai berikut: Suatu sistem yang mempunyai kelas untuk mempresentasikan kodi dalam diskret yang didalamnya aplikasi yang suda dimodelkan. Yang bersifat fisika, bisnis dan logika. Diagram

Class merupakan konsep yang dikenali dalam terminalogi ilmu komputer yang memiliki metode pengurutan dan pencarian atau sesuatu yang bersifat perilaku.

II.12.4. Activity Diagram

Menurut Adi Nugroho (2010 : 6-7) Diagram aktifitas (*Activity Diagram*) merupakan bentuk khusus dari *State machine* yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja terjadi dalam sistem atau perangkat lunak yang sedang dikembangkan, suatu diagram aktivitas memuat didalamnya *activity state* merepresentasikan eksekusi pernyataan dalam suatu prosedur atau kinerja suatu aktifitas dalam suatu aliran kerja. Dan diagram aktivitas dapat memperlihatkan aliran nilai-nilai objek, seperti layaknya aliran-aliran kendali.

Tabel II.5 Komponen Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Grady Booch ; 2013)

Kesimpulan *activity diagram* adalah sebagai berikut: *activity diagram* merupakan bentuk kusus dari *state machine* yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja terjadi dalam sistem atau perangkat lunak yang dikembangkan dan dapat memperlihatkan aliran nilai-nilai objek. *activity diagram* sistem yang menggambarkan alir aktivitas dalam sistem yang sedang dirancang. Dan merepresentasikan eksekusi pernyataan dalam suatu prosedur atau kinerja suatu aktifitas dalam suatu aliran kerja.

II.13. Normalisasi

Menurut Kusri (2011 : 39-43) Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Sedangkan

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari E-R ke bentuk fisik.

Dalam perpektif normalisasi sebuah database dikatakan baik jika setiap tabel yang membentuk basis data sudah berada dalam keadaan normal. Suatu tabel dikatakan normal jika :

- a. Jika ada dekomposisi/penguraian tabel, maka dekomposisinya dijamin aman (*lossless-join decomposition*)
- b. Terpeliharanya ketergantungan functional pada saat perubahan data (*dependency preservation*)
- c. Tidak melanggar *Boyce Code Normal Form* (BCNF), jika tidak bisa minimal tidak melanggar bentuk normalisasi ketiga

Yang dimaksud dengan ketergantungan fungsional (*functional dependency*) adalah :

Diberikan sebuah tabel/relasi T, atribut B dari T bergantung secara fungsi pada atribut A dari T jika dan hanya jika setiap nilai B dari T punya hubungan dengan tepat satu nilai A dalam T (dalam setiap satu waktu).

Beberapa kondisi yang diujikan pada proses normalisasi :

- a. Menambah data/insert
- b. Mengedit/update
- c. Menghapus/delete
- d. Membaca/retrieve

II.13.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.

b. Bentuk normal tahap pertama (1NF)

Sebuah table disebut 1NF jika memenuhi syarat-syarat sebagai berikut :

1. Tidak ada baris yang duplikasi dalam tabel tersebut
2. Masing-masing cell bernilai tunggal

Catatan : permintaan yang menyatakan tidak ada baris yang duplikasi dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut merupakan kombinasi dari semua kolom.

Contoh :

- Tabel kuliah (kode_kul, nama_kul, sks, semester, nama_dos)
- Tabel jadwal (kode_kul, waktu, ruang)

c. Bentuk normal tahap kedua (2NF)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*).

Contoh :

Misal tabel nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel nilai, misalnya kita tambahkan sebuah atribut yang bersifat redundan, yaitu nama_mhs, maka tabel nilai ini dianggap melanggar 2NF. *Primary key* pada tabel nilai adalah (kode_kul, nim). Penambahan atribut baru (nama_mhs) akan menyebabkan adanya ketergantungan

fungsiional yang baru yaitu $nim > nama_mhs$. Karena atribut $nama_mhs$ ini hanya memiliki ketergantungan parsial pada *primary key* secara utuh (hanya tergantung pada nim , padahal nim hanya bagian dari *primary key*).

Bentuk normal kedua ini dianggap belum memadai karena meninjau sifat ketergantungan atribut terhadap *primary key*.

d. Bentuk normal tahap ketiga (3NF)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsiional dengan relasi $X > A$, dimana A mewakili semua atribut unggal di dalam tabel yang tidak ada di dalam X , maka :

- X haruslah superkey pada tabel tersebut
- Atau A merupakan bagian dari *primary key* pada tabel tersebut

Misalkan pada tabel mahasiswa, atribut $alamat_mhs$ dipecah kedalam $alamat_jalan$, $alamat_kota$ dan $kode_pos$. Bentuk ini tidak memenuhi 3NF, karena terdapat ketergantungan fungsiional baru yang muncul pada tabel tersebut, yaitu :

$alamat_jalan\ nama_kota > kode_pos$

dalam hal ini ($alamat_jalan$, $nama_kota$) bukan superkey sementara $kode_pos$ juga bukan bagian dari *primary key* pada tabel mahasiswa. Jika tabel mahasiswa didekomposisi menjadi tabel mahasiswa dan tabel alamat, maka telah memenuhi 3NF. Hal itu dapat dibuktikan dengan memeriksa dua ketergantungan fungsiional pada tabel alamat tersebut, yaitu :

alamat_jalan nama_kota > kode_pos

kode_pos > nama_kota

ketergantungan fungsional yang pertama tidak melanggar 3NF, karena (alamat_jalan, nama_kota) merupakan superkey (sekali sebagai primary key) dari tabel alamat tersebut. Demikian juga dengan ketergantungan fungsional yang kedua meskipun (kode_pos) bukan merupakan superkey, tetapi nama_kota merupakan bagian dari primary key dari tabel alamat. Karena telah memenuhi 3NF, maka tabel tersebut tidak perlu didekomposisi lagi.

e. Bentuk normal tahap keempat dan kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk 4NF berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk 5NF merupakan nama lain dari *project join normal form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1NF
- Relasi harus bergantung fungsi pada atribut superkey.

Kesimpulan Normalisasi adalah sebagai berikut: untuk memperbaiki /membangun data dengan model data relasional dan secara umum lebih tepat dikoneksikan dengan data model logika. Suatu pengelompokan data

kedalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan mereka, sehingga terwujud satu bentuk basis data yang mudah untuk dimodifikasi