

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan (Asbon Hendra, 2012:157).

II.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*) (Hartono, 2004).

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan atau sasaran. Unsur-unsur yang saling melengkapi tersebut terdapat di dalam sistem yang disebut dengan nama subsistem. Subsistem-subsistem tersebut harus selalu berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien.

II.2. Data

Data adalah detugas tentang benda, kejadian, aktifitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh langsung kepada pemakai. Data dapat berupa nilai terformat, teks, citra, audio, dan video (Jogianto,2004).

Data yang terformat adalah data dengan suatu format tertentu. Misalnya, data yang menyatakan tanggal atau jam, atau menyatakan nilai mata uang.

Teks adalah sederhana huruf, angka, dan simbol-simbol khususnya (misalnya “+” dan “\$”) yang kombinasinya tidak tergantung pada masing-masing item secara individual. Contoh teks adalah koran.

Citra (*images*) adalah data dalam bentuk gambar. Citra dapat berupa grafik, foto, hasil rontgen, dan tanda tangan ataupun gambar yang lain.

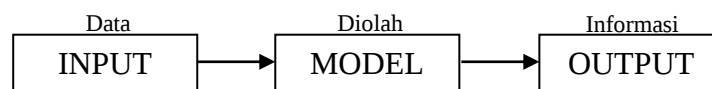
Audio adalah data dalam bentuk suara. Instrumen musik, suara orang atau suara binatang, gemercik air, detak jantung merupakan beberapa contoh data audio.

Video menyatakan data dalam bentuk sejumlah gambar yang bergerak dan bisa saja dilengkapi dengan suara. Video dapat digunakan untuk mengabadikan suatu kejadian atau aktifitas.

II.3. Informasi

Informasi adalah suatu data yang telah diproses sehingga dapat mengurangi ketidak jelasan tentang keadaan atau suatu kejadian. Sedangkan kata data itu sendiri adalah fakta atau kenyataan yang sebenarnya.

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses tranformasi data menjadi suatu informasi (input-proses-output). Sedangkan data adalah *raw material* untuk suatu informasi (Asbon Hendra, 2012:167).



Gambar II.1 Siklus Pengolahan Data

(Sumber : Riyanto, et all ; 2009)

II.3.1. Ciri-ciri Informasi

Dalam lingkup sistem informasi, informasi memiliki ciri-ciri seperti yang dijelaskandi bawah ini : (Riyanto, et all, 2009)

1. Benar atau salah, ini dapat berhubungan dengan realitas atau tidak. Bila penerima informasi yang salah mempercayainya, akibatnya sama seperti yang benar.
2. Baru, informasi dapat sama sekali baru dan segar bagi penerimanya.
3. Tambahan, informasi dapat memperbaharui atau memberikan tambahan pada informasi yang telah ada.
4. Korektif, informasi data menjadi suatu korektif atas salah satu informasi sebelumnya.

5. Penegas, informasi dapat mempertegas informasi yang telah ada. Ini masih berguna karena meningkatkan persepsi penerimanya atas kebenaran informasi tersebut.

II.4. Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang , fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian *internal* dan *eksternal* yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdas (Hartono,2004).

II.4.1 Klasifikasi Sistem Informasi

Klasifikasi pengelompokkan sistem informasi (Jogianto,2004) berdasarkan pada :

1. Sistem Informasi Menurut Level Organisasi : Sistem informasi departement, sistem informasi perusahaan, dan sistem informasi antar organisasi.
2. Sistem Informasi Fungsional : sistem informasi akuntansi, sistem informasi keuangan, sistem informasi manufaktur, sistem informasi pemasaran, dan sistem informasi sumber daya manusia.

3. Sistem Informasi Berdasarkan Dukungan Yang Tersedia : sistem pemrosesan transaksi (TPS), sistem informasi manajemen (MIS), sistem pendukung keputusan (DSS), sistem informasi eksekutif (EIS), sistem pendukung kelompok (GSS), dan sistem pendukung cerdas (ESS).
4. Sistem Informasi Menurut Aktivitas Manajemen : sistem informasi pengetahuan, sistem informasi operasional, sistem informasi manajerial, dan sistem informasi strategis.
5. Sistem Informasi Menurut Arsitektur Sistem : sistem berbasis mainframe, sistem komputer pribadi (PC) tunggal, dan sistem komputasi jaringan.
6. Sistem Informasi Geografis
7. Sistem ERP (*Enterprise Resource Planning*).

II.5. Geografi

Geografi adalah ilmu yang mempelajari permukaan bumi dengan menggunakan pendekatan keruangan, ekologi, dan kompleks wilayah. Fenomena yang diamati merupakan dinamika perkembangan dan pembangunan wilayah yang ada dalam keseharian, misalnya informasi mengenai letak dan persebaran dari kejadian-kejadian alamiah maupun fenomena terdapatnya sumberdaya. Ketersediaan data yang bersifat geografi, dimana yang memiliki atribut utama keruangan akan memudahkan banyak kepentingan.

ESRI (*Environmental System Research Institute*) mendefinisikan Sistem Informasi Geografis adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografis dan personil yang dirancang secara

efisien untuk memperoleh, menyimpan, memperbaharui, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografi (Riyanto, et al, 2009:35-36).

II.6. Sistem Informasi Geografis

Sistem informasi geografis adalah sebuah rangkaian sistem yang memanfaatkan teknologi digital untuk melakukan analisa spasial. Sistem ini memanfaatkan perangkat keras dan lunak komputer untuk melakukan pengolahan data. (Eko Budyanto,2004)

II.6.1. RuangLingkupSistemInformasiGeografis(SIG)

Berikut merupakan beberapa ruang lingkup dari Sistem Informasi Geografis antara lain adalah ;

1. Input

Pada tahap input (pemasukan data) yang dilakukan adalah mengumpulkan dan mempersiapkan data spasial atau atribut dari berbagai sumber data. Data yang digunakan harus dikonversikan menjadi format digital yang sesuai, proses konversi yang dilakukan dikenal dengan proses dijitasi (*digitizing*).

2. Manipulasi

Manipulasi data merupakan proses *editing* terhadap data yang telah masuk, hal ini dilakukan untuk menyesuaikan tipe dan jenis data agar sesuai dengan sistem yang akan dibuat, seperti : penyamaan skala, pengubahan sistem proyeksi, generalisasi dan sebagainya.

3. Manajemen Data

Tahap ini meliputi seluruh aktifitas yang berhubungan dengan pengolahan data (menyimpan, mengorganisasi, mengelola, dan menganalisis data) kedalam sistem penyimpanan permanen, seperti : sistem *file server* atau *database server* sesuai kebutuhan sistem. Jika menggunakan sistem *file server*, data disimpan dalam bentuk **.txt*, **.dat*, dan lain-lain. Sedangkan jika menggunakan sistem *database server*, biasanya memanfaatkan *software Database Management System (DBMS)*, seperti *MySQL*, *SQL Server*, *ORACLE*, dan *DBMS* sejenis lainnya.

4. Query

Suatu metode pencarian informasi untuk menjawab pertanyaan yang diajukan oleh pengguna sistem pada SIG dengan sistem *file server*, *query* dapat dimanfaatkan dengan bantuan *compiler* atau *interpreter* yang digunakan dalam mengembangkan sistem, sedangkan untuk SIG dengan sistem *database server*, dapat memanfaatkan *SQL (structured query language)* yang terdapat pada *DBMS* yang digunakan.

5. Analisis

Terdapat dua jenis fungsi analisis dalam SIG, yaitu fungsi analisis spasial dan analisis atribut. Fungsi analisis spasial adalah operasi yang dilakukan pada data spasial. Sedangkan fungsi analisis atribut adalah fungsi pengolahan data atribut yaitu data yang tidak berhubungan dengan ruang.

6. Visualisasi (Data Output)

Penyajian hasil berupa informasi baru atau *database* yang ada baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti dalam bentuk peta (atribut peta dan atribut data), table, grafik dan lain-lain (Riyanto, et al, 2009:38-39).

II.6.2. Manfaat Sistem Informasi Geografis

Dari dunia nyata diambil tiga hal penting yaitu posisi dan klasifikasi, atribut, serta hubungan antaritem. Ketiga hal tersebut diolah sebagai dasar analisa sistem spasial dalam SIG. Dengan dasar tersebut akan dapat diperoleh manfaat dari SIG sebagai berikut :

1. Menjelaskan tentang lokasi atau letak
2. Menjelaskan kondisi ruang
3. Menjelaskan suatu kecenderungan (*trend*)
4. Menjelaskan tentang pola spasial (*spacial patern*)
5. Pemodelan

II.6.3. Cara Kerja SIG

SIG dapat menyajikan *real world* (dunia nyata) pada monitor sebagaimana lembaran peta dapat merepresentasikan dunia nyata di atas kertas. Tetapi, SIG memiliki kekuatan lebih dan fleksibilitas dari pada lembaran pada kertas. Peta merupakan representasi grafis dari dunia nyata, objek-objek yang dipresentasikan di atas peta disebut unsur peta atau *map features*(contohnya adalah sungai, taman, kebun, jalan dan lain-lain). Karena peta mengorganisasikan unsur-unsur

berdasarkan lokasi-lokasinya. SIG menyimpan semua informasi dekskriptif unsur-unsurnya sebagai atribut-atribut didalam basis data. Kemudian, SIG membentuk dan menyimpannya didalam tabel-tabel (*relasional*) dengan demikian, atribut-atribut ini dapat diakses melalui lokasi-lokasi unsur-unsur peta dan sebaliknya, unsur-unsur peta juga dapat diakses melalui atribut-atributnya.(Eddy Prahasta, 2005)

II.7. Peta

Peta merupakan penyajian grafis dari permukaan bumi dalam skala tertentu dan digambarkan pada bidang datar melalui sistem proyeksi peta dengan menggunakan simbol-simbol tertentu sebagai perwakilan dari objek-objek spasial di permukaan bumi. Peta ini juga memungkinkan perubahan skala, animasi gabungan, gambar, suara, dan bisa terhubung ke sumber informasi tambahan melalui internet. Sebuah peta harus memiliki syarat-syarat sebagai berikut :

1. Peta tidak boleh membingungkan. Dalam hal ini peta perlu dilengkapi :
 - a. Keterangan atau legenda (*legend*)
 - b. Skala (*Scale*) peta
 - c. Judul Peta (apa isinya)
 - d. Bagian dunia mana
2. Peta harus dengan mudah dapat dimengerti atau ditangkap maknanya oleh si pemakai peta. Supaya mudah dimengerti atau ditangkap maknanya, digunakan :
 - a. Tata warna

- b. Simbol (terutama pada peta tematik)
 - c. Sistem proyeksi dan sistem koordinat
3. Peta harus memberikan gambaran yang sebenarnya, yang berarti peta harus cukup teliti sesuai dengan tujuannya (Riyanto, et al, 2009:3-4).

II.8. MapInfo

MapInfo adalah *software* pengolah data spasial yang banyak digunakan dalam analisis sistem informasi geografis. Software ini memiliki kemampuan seperti *software-software* pengolah spasial lain seperti Arc Info atau Arc view. MapInfo merupakan *software* pengolah data spasial yang terpadu dengan tabel. Melalui software MapInfo operator dapat membuat, menampilkan, serta mengadakan perubahan terhadap data spasial atau peta. (Eko Budianto, 2004)

II.9. Basis Data

Basis data (bahasa Inggris : *database*) adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Tempat penyimpanan utama sebuah *database* dinamakan dengan tabel. Pada umumnya, tabel-tabel akan dikelompokkan dalam sebuah *database* (Riyanto, 2009).

Basis Data atau *database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. Untuk mendefinisikan, mengatur, dan

memperoses database, DBMS (*Database Management System*) yang akan digunakan adalah *MySQL*.

MySQL merupakan software RDBMS (atau server database) yang dapat mengelola database dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (multi-user), dan dapat melakukan suatu proses secara sinkron atau berbarengan (multi-threaded).

Lisensi *MySQL* terbagi menjadi dua, yaitu *MySQL* sebagai produk *open source* dibawah GNU (*General Public Licence*) yang gratis dan lisensi dari versi komersialnya yang memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis (Budi Raharjo, 2011:22).

II.9.1. Hierarki Data dalam Database

Data dalam sebuah *database* disusun berdasarkan sistem hierarki yang unik, yaitu :

1. *Database*, merupakan kumpulan *file* yang saling terkait satu sama lain, misalnya *file* data induk karyawan, *file* jabatan, *file* penggajian dan lain sebagainya. Kumpulan *file* yang tidak saling terkait satu sama lain tidak dapat disebut *database*.
2. *File*, yaitu kumpulan dari *record* yang saling terkait dan memiliki format *field* yang sama dan sejenis.
3. *Record*, yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.
4. *Field*, yaitu atribut dari *record* yang menunjukkan suatu item dari data seperti nama, alamat, dan lain sebagainya.

5. *Byte*, yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.
6. *Bit*, yaitu bagian terkecil dari data secara keseluruhan yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte* (Budi Sutedjo Dharma Oetomo, 2006:102).

II.10. Entity Relationship Diagram (ERD)

Database adalah kumpulan file yang saling berkaitan. Pada model data relational, hubungan antar *file* direlasikan dengan kunci relasi (*relation key*) yang merupakan kunci utama dari masing-masing *file*. Relasi antara 2 *file* atau 2 tabel dapat dikategorikan menjadi 3 macam, adalah sebagai berikut :

1. One to onerelationship dua file

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Seperti pada pelajaran privat di mana satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan table dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.

2. One to manyrelationship dua file

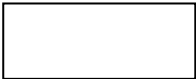
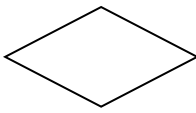
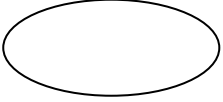
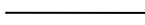
Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak lawan satu. Seperti

pada sistem pengajaran di sekolah dasar, di mana satu guru mengajar banyak siswa dan siswa hanya diajar oleh satu guru.

3. *Many to many relationship* dua file

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi, di mana satu dosen mengajar banyak mahasiswa dan mahasiswa diajar oleh banyak dosen (Tata Sutabri, 2004:208-209).

Tabel II.1 : Simbol – Simbol pada ERD

Notasi	Nama	Fungsi
	Entitas	Entitas adalah suatu objek yang dapat didefinisikan dalam lingkungan pemakai.
	Relasi	Relasi menunjukkan adanya hubungan diantara sejumlah entitas.
	Atribut	Atribut berfungsi mendeskripsikan karakter entitas.
	Garis	Garis sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

(Sumber : Janner Simarmata dan Imam Prayudi ; 2006)

II.10.1. Teknik Normalisasi

Normalisasi adalah proses langkah-langkah yang fleksibel untuk menggantikan hubungan yang ditentukan oleh koleksi yang berurutan dimana hubungan mempunyai suatu struktur yang lebih regular dan sederhana.

Adapun tahap-tahap normalisasi yaitu :

1. Bentuk Normal Pertama (*First Normal Form / 1NF*)

Bentuk normal pertama mempunyai ciri, yaitu setiap data dibentuk dalam *flat file*, data dibentuk dalam satu *record* dan nilai dari *field-field* yang menyatakan data tunggal dan bukan gabungan dari nilai-nilai. Tidak ada *set* atribut yang berulang-ulang atau atribut bernilai ganda (*multivalue*).

2. Bentuk Normal Kedua (*Second Normal Form / 2NF*)

Bentuk normal kedua mempunyai syarat, yaitu bentuk data memenuhi kriteria bentuk pertama. Atribut bukan kunci haruslah bergantung secara fungsional pada kunci utama (*primary key*), sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci *field*. Kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

3. Bentuk Normal Ketiga (*Third Normal Form / 3NF*)

Untuk menjadi bentuk ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak mempunyai hubungan transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan *primary key* secara menyeluruh.

4. Bentuk Normal *Boyce-Codd*(*Boyce-Codd Normal Form/BCNF*)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk

normal pertama dan setiap atribut bergantung fungsi pada atribut *superkey*(Riyanto, et al, 2009:31).

II.10.2. Kamus Data

Kamus data ikut berperan dalam perancangan dan pembangunan sistem informasi karena peralatan ini berfungsi untuk :

1. Menjelaskan arti aliran data dan penyimpanan dalam penggambaran dalam data *flow diagram*.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran, misalnya data alamat diurai menjadi jalan, nomor, kota, Negara dan kode pos.
3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut (Budi Sutedjo Dharma Oetomo, 2006:118).

Tabel II.2. Simbol-simbol dalam Kamus Data

Notasi	Uraian
=	Terdiri atas, mendefinisikan, diuraikan menjadi, artinya contoh : nama=sebutan+nama1+nama2+gelar1+gelar2
+	Dan
()	Optional (pilihan boleh ada atau boleh tidak) contoh: alamat=alamat rumah+(alamat surat)
{ }	Pengulangan contoh nama1={karakter_valid}
[]	Memilih salah satu dari sejumlah alternative, seleksi contoh: sebutan=[Bapak Ibu Yang Mulia]
**	Komentar contoh: *seminar yang akan diikuti*

	Pemisah sejumlah alternative pilihan antara symbol []
--	--

(Sumber : Budi Sutedjo Dharma Oetomo ;2006)

II.11. *Unified Modeling Language (UML)*

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

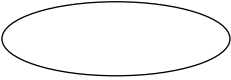
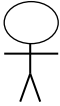
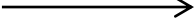
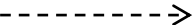
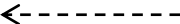
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram* yaitu :

Tabel II.3. Simbol Use Case

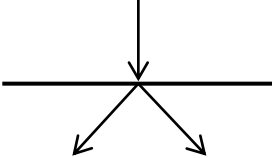
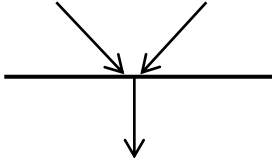
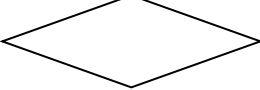
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva, et al ; 2015)

1. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

Tabel II.4. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva, et al ; 2015)

2. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel dibawah ini :

Tabel II.5. *Multiplicity Class Diagram*

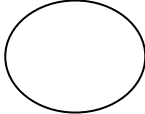
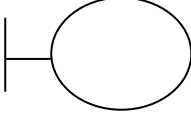
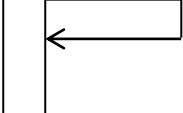
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva, et al ; 2015)

1. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

Tabel II.6. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva, et al ; 2015)

II.12. MySQL

MySQL merupakan *software* yang tergolong *database server* dan bersifat *open source*. *open source* menyatakan bahwa *software* ini di lengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal menarik lainnya adalah MySQL juga bersifat multiplatform. MySQL dapat dijalankan pada berbagai sistem operasi. (Abdul Kadir, 2009).

Pengaksesan data dalam *database* dapat dilakukan dengan mudah melalui SQL (*Structured Query Language*). Data dalam *database* bisa di akses mealalui aplikasi non-Web (misalnya dengan Visual Basic) maupun aplikasi Web (misalnya dengan PHP).

II.13. Web Browser

Web Browser adalah suatu perangkat lunak yang digunakan untuk menampilkan halaman-halaman website yang berada di internet.

Terdapat beberapa macam web browser yang dapat kita pakai untuk menampilkan halaman-halaman website. Ada 3 jenis *web browser* yang sering dipakai adalah :

1. Internet Expoler

2. Netscape
3. Mozilla

II.14. Web Server

Web server adalah software yang menjadi tulang belakang dari *world wide web*(www). *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti Netscape Navigator, Internet Explorer, Mozilla, dan program browser lainnya. (Febrian, Jack, 2004). Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar, disebut dengan format SGML (*standar general markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browsers* sesuai dengan kemampuan browser tersebut. Contohnya, bila data yang dikirim berupa gambar, *browser* yang hanya mampu menampilkan teks (misalnya lynx) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja. *Web server*, untuk berkomunikasi dengan client-nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*).

II.15. WWW (World Wide Web)

Web atau *World Wide Web* (WWW) adalah salah satu fasilitas yang disediakan di internet. Web merupakan dunia maya di internet yang terdapat

ribuan info tentang segala hal dan dikemas dalam bentuk dokumen *hypertext* (Nugroho, 2005). *Hypertext* merupakan konsep dasar dari *world wide web*. Dokumen *hypertext* adalah salah satu dokumen yang memungkinkan untuk menjelajahi dari suatu halaman web ke halaman web yang lainnya dengan menggunakan suatu *links* (menghubungkan)

Web bekerja dalam jaringan komputer yang memanfaatkan teknologi *Hypertext Transfer Protocol*(HTTP).*Uniform Resource Locator*(URL), *Protocol Transfer* dan *Domain Name Sistem* (DNS).

II.16. Bahasa Pemrograman (*Program Language*)

Bahasa pemrograman atau *Program Language* sangat diperlukan untuk melakukan perintah menjalankan suatu instruksi tertentu, dalam sistem yang akan di rancang ini bahasa pemrograman yang digunakan adalah *HTML*, *PHP* dan *JavaScript*. Sedangkan editor yang digunakan adalah Dreamweaver.

1. HTML

HTML atau (*Hypertext Markup Language*) adalah bahasa yang dipakai untuk menampilkan informasi dalam bentuk *hypertext*. *HTML* juga mendukung bagaimana suatu informasi ditampilkan. Sesuai dengan namanya, bahasa ini menggunakan markup untuk menandai perintah-perintahnya.

HTML terdiri atas elemen dan atribut dari elemen, dimana elemen dan atribut tersebut digunakan untuk mengatur tampilan web. Istilah untuk

menyebut elemen HTML dapat juga disebut sebagai tag (Riyanto, et al, 2009:247).

Dokumen HTML yang paling sederhana adalah sebagai berikut :

```
<HTML>
<HEAD>
<TITLE>Judul halaman web yang ditulis</TITLE>
</HEAD>
  <Body>
    Isi dokumen HTML
  </Body>
</HTML>
```

Gambar II.2.Listing HTML

(Sumber : Riyanto, et al ; 2009)

2. PHP

PHP merupakan perangkat lunak yang digunakan untuk melakukan interpretasi dari kode PHP menjadi kode HTML sehingga hasilnya dapat ditampilkan didalam *web browser* (Budi Raharjo, 2011:246).

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 PHP, prosesor itu akan menerjemahkan dan mengeksekusi semua perintah dalam halaman tersebut, dan kemudian menampilkan hasilnya ke *browser* sebagai halaman HTML biasa. PHP dapat ditanamkan langsung kedalam HTML. Kode PHP dipisahkan dari HTML dengan menggunakan tanda *Start* dan *End*. Ketika sebuah dokumen dibaca prosesor PHP hanya

menerjemahkan area yang ditandai saja, dan menampilkan hasilnya pada tempat yang sama (Adhi Prasetyo, 2014:122-123).

```
<html>
<head>
  <title>contoh halaman php</title>
</head>
<body>
  <font color = "blue"> PHP kode saya akan membuat
  halaman ini menampilkan</font>
  <?php
  Print ("Anda berada disitus Prothelon!");
  ?>
</body>
</html>
```

Gambar II.3 Kode PHP

(Sumber : Adhi Prasetyo ; 2014)

3. *JavaScript*

JavaScript merupakan bahasa pemrograman *web client side*. Kalau HTML digunakan untuk membuat halaman web statis, maka *JavaScript* dapat digunakan untuk membuat aplikasi matematis, efek animasi sederhana, bahkan juga untuk membuat *game* (Riyanto, et al, 2009:266). *JavaScript* adalah bahasa pemrograman yang digunakan untuk membuat web lebih dinamis dan interaktif. *JavaScript* terintegrasi langsung dengan html. Kode *JavaScript* biasanya dituliskan dalam bentuk fungsi yang ditaruh di tag *<head>* yang dibuka dengan tag *<Script type="teks/javascript">*. Gambar berikut merupakan contoh dari kode *JavaScript*, yaitu :

```
<script type= "teks/javascript">  
    alert ("Hello Dunia !");  
</script>
```

Gambar II.4 Kode JavaScript

(Sumber : Adhi Prasetyo ; 2014)

Kode *JavaScript* juga dapat diletakkan di file tersendiri yang berekstensi *js* singkatan dari *JavaScript*. Untuk memanggil kode *JavaScript* yang terdapat di *file* sendiri. Dibagian awal *<head>* harus ditentukan dahulu nama *file.js* yang dimaksud dengan menggunakan contoh kode seperti berikut :

```
<script type="teks/javascript" src="alamat.js"></script>
```

Gambar II.5 Kode JavaScript

(Sumber : Adhi Prasetyo ; 2014)

Sidik (2011:4) menjelaskan cara kerja *JavaScript* secara singkat sebagai berikut :

“*JavaScript* adalah program dalam bentuk *script*, yang dijalankan oleh *interpreter* yang telah ditanamkan kedalam *browser web*, sehingga *browser web* dapat mengeksekusi program *JavaScript*” (Adhi Prasetyo, 2014:291-292).