

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Dasar Sistem

Konsep dasar sistem akan menguraikan beberapa pengertian sistem, karakteristik sistem, pengertian dan komponen sistem informasi.

II.1.1. Pengertian Sistem

Merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Asbon Hendra (2012:157).

Menurut Jerry FithGerald (2012 : 157), menyatakan bahwa sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Menurut Ludwig Von Bertalanfy (2012 : 157), menyatakan bahwa sistem adalah merupakan seperangkat unsur yang saling terkait dalam suatu antarrelasi di antara unsur-unsur tersebut dengan lingkungan.

Menurut Anatol Rapoport (2012 : 157), menyatakan bahwa sistem adalah merupakan suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

Menurut L.Ackof (2012 : 158), menyatakan bahwa sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Dari beberapa pengertian diatas diambil kesimpulan bahwa sistem adalah kumpulan atau kelompok yang saling terkait untuk mencapai tujuan.

II.1.2. Karakteristik Sistem

Menurut Asbon Hendra (2012:158), suatu sistem mempunyai karakteristik tertentu, yaitu:

1. Komponen (*Component*)

Suatu sistem dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem, tidak peduli betapa pun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar disebut *supra sistem*. Sebagai contoh, suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem.

2. Batas sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.3. Elemen sistem

Menurut Hanif Al Fatha (2012:160), Ada beberapa elemen yang membentuk suatu sistem, yaitu:

1. Tujuan, merupakan tujuan dari sistem tersebut yang dapat berupa tujuan usaha, kebutuhan, masalah dan prosedur pencapaian tujuan.

2. Batasan, merupakan batasan-batasan yang ada dalam mencapai tujuan dari sistem, dimana batasan ini berupa peraturan-peraturan, biaya-biaya, dan peralatan.
3. Kontrol, merupakan pengawas dari pencapaian tujuan sistem, yang dapat berupa kontrol pemasukan data (*input*), kontrol pengeluaran data (*output*) dan kontrol pengoperasian.
4. Input, merupakan bagian dari sistem yang bertugas untuk menerima data masukan, dimana data dapat berupa masukan, frekuensi pemasukan data, dan jenis pemasukan data.
5. Proses, merupakan bagian yang memproses masukan data menjadi informasi sesuai dengan keinginan penerima, dimana proses dapat klasifikasi, peringkasan dan pencarian.
6. Output, merupakan keluaran atau tujuan akhir dari sistem, output dapat berupa laporan, dan grafik.
7. Umpan balik, merupakan elemen sistem yang mempunyai tugas untuk melihat kembali apakah sistem telah berjalan sesuai dengan yang diinginkan, umpan balik dapat berupa perbaikan, pemeliharaan.

II.2. Pengertian Informasi

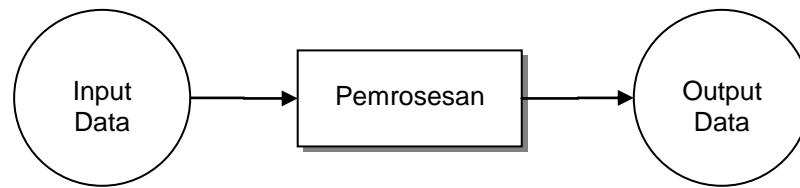
Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti

bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi=input-proses-output. Sementara data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relatif, tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya atau sebaliknya.

II.2.1. Sistem Informasi

Menurut Robert A Leitch (2012 : 169), Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Akhirnya Sistem Informasi Manajemen (SIM) dapat didefinisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya (*Kertahadi, 1995*). Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan, dan menyajikan sinergi organisasi pada proses (*Murdick dan Ross, 1993*). Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output – IPO*) dapat dilihat pada gambar berikut :



Gambar II.1. Konsep sistem informasi

Sumber : Asbon Hendra (2012 : 168)

II.2.2. Komponen Sistem Informasi

Kusrini Mkom dan Andri kuniyo, Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*Building Block*) yaitu Blok Masukan (*Input Block*), Blok Model (*Model Block*), Blok Keluaran (*Output Block*), Blok Teknologi (*Technologi Block*), Blok Dasar Data (*Database Block*) dan Blok Kendali (*Control Block*).

Sebagai sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai suatu sasaran.

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang dimasukkan, yang dapat berupa dokumen-dokumen.

2. Blok model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Tecnologi Block*)

Teknologi merupakan “Kotak Tool” dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model menyimpan dan mengakses data, menghubungkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari bagian utama yaitu:

- a. Teknisi (*Humanware* dan *Brainware*)
- b. Perangkat Lunak (*Software*)
- c. Perangkat keras (*Hardware*)

Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membantunya untuk dapat beroperasi.

5. Blok Basis Data (*Database Block*)

Basis data merupakan komponen dari data yang saling terhubung satu dengan yang lainnya, tersimpan di perangkat lunak untuk memanipulasinya. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas.

6. Blok Kendali (*Control Block*)

Agar sistem dapat berjalan sesuai dengan yang diinginkan maka perlu diterapkan pengendalian-pengendalian di dalamnya. Banyak hal yang dapat

merusak sistem informasi seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan sistem itu sendiri, dan lain-lain.

II.2.3. Jenis Sistem Informasi

Menurut Ida Nuraida (2008:26), Sistem informasi memiliki beberapa jenis yaitu:

1. Sistem informasi menurut level organisasi

Berdasarkan level organisasi, sistem informasi dikelompokkan menjadi beberapa bagian, yaitu:

- a. Sistem informasi Departemen
- b. Sistem informasi perusahaan
- c. Sistem informasi antarorganisasi.

2. Sistem informasi fungsional

Sistem informasi fungsional adalah sistem informasi yang ditujukan untuk memberikan informasi bagi kelompok orang yang berada pada bagian tertentu dalam perusahaan. Beberapa informasi fungsional yang umum yaitu:

- a. Sistem informasi akuntansi
- b. Sistem informasi keuangan,
- c. Sistem informasi manufaktur
- d. Sistem informasi pemasaran,
- e. Sistem informasi SDM.

3. Sistem informasi berdasarkan dukungan yang tersedia

Berdasarkan dukungan yang diberikan pada pemakai, sistem informasi yang digunakan semua area fungsional dapat diklasifikasikan yaitu sistem

pemrosesan transaksi, sistem informasi manajemen, sistem informasi perkantoran, sistem pendukung keputusan, sistem informasi eksekutif, sistem pendukung kelompok, dan sistem pendukung cerdas.

4. Sistem Informasi Geografis

Sistem informasi geografis adalah sistem berbasis komputer yang digunakan untuk menyimpan dan manipulasi informasi geografis. Sistem informasi geografis dapat menggabungkan tugas-tugas pengambilan keputusan seperti mencari jarak terdekat atau tercepat dari posisi A ke posisi B, menentukan/memilih lokasi lain memiliki pola serupa, mengelompokkan daerah perindustrian untuk meminimalkan jarak perjalanan.

II.2.4. Tujuan Sistem Informasi

Adapun tujuan sistem informasi adalah untuk menghasilkan informasi yang baik dan lebih berguna untuk mempermudah dan mengefisienkan waktu, biaya dalam mengambil suatu keputusan dan untuk mengurangi ketidakpastian bagi pengambil keputusan. Adapun tujuan utama yang umum dari sistem informasi adalah:

1. Untuk mendukung fungsi kepengurusan manajemen
2. Untuk mendukung pengambilan keputusan manajemen
3. Untuk mendukung kegiatan operasi perusahaan.

II.3. Sistem Informasi Geografis (*Geographic Information System*)

II.3.1. Pengertian GIS

Menurut Eddy Prahasta (2009 :109), pada dasarnya istilah sistem informasi geografis merupakan gabungan dari tiga unsur pokok: sistem, informasi, dan geografis. Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG. Dengan melihat unsur-unsur pokoknya, maka sudah jelas bahwa SIG juga merupakan salah satu tipe sistem informasi seperti yang telah dibahas di muka; tetapi dengan tambahan unsur “geografis”. Atau SIG merupakan suatu sistem yang menekankan pada unsur “Informasi Geografis”.

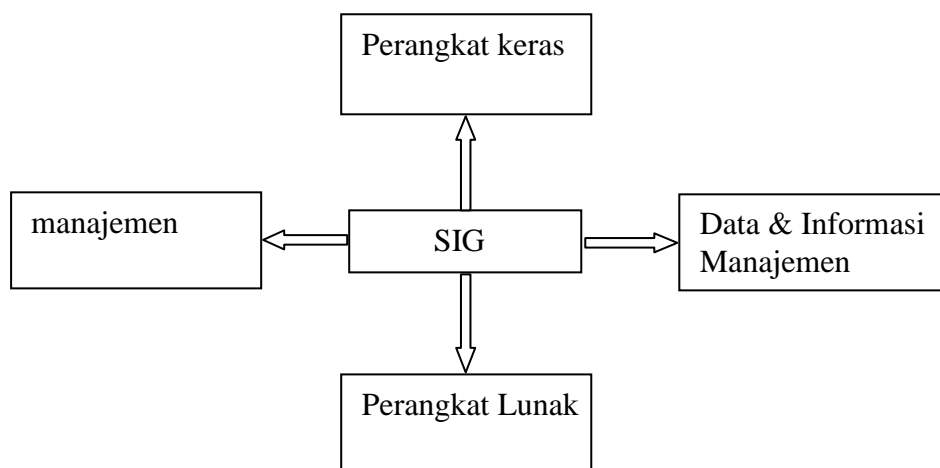
Istilah “Geografis” merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau bahkan tertukar satu sama lainnya hingga muncullah istilah yang ketiga. Geospasial. Ketiga istilah ini mengandung pengertian yang kurang lebih serupa didalam konteks SIG. Penggunaan kata “Geografis” mengandung pengertian suatu persoalan atau hal mengenai (wilayah dipermukaan) bumi: baik permukaan dua dimensi atau tiga dimensi. Dengan demikian, istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, atau informasi mengenai keterangan-keterangan (atribut) objek penting yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui.

Dengan memperhatikan pengertian sistem informasi diatas, maka SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber

daya fisik dan logika yang berkenaan dengan objek-objek penting yang terdapat dipermukaan bumi.

II.3.2. Komponen GIS

Komponen kunci dalam GIS adalah sistem komputer, data geospasial (data atribut) dan pengguna, yang dapat digambarkan sebagai berikut :



Gambar II.2. Komponen kunci SIG

Sumber : Eddy Prahasta (2009:121)

Sistem komputer GIS terdiri dari perangkat keras (hardware), perangkat lunak (software) dan prosedur untuk penyusunan pemasukkan data, pengolahan, analisis, pemodelan (modelling), dan penayangan data geospasial.

Sumber-sumber data geospasial adalah peta digital, foto udara, citra satelit, tabel statistik dan dokumen lain yang berhubungan. Data geospasial dibedakan menjadi:

- a. Data grafis/geometris, mempunyai tiga elemen : titik (node), garis (arc) dan luasan (poligon) dalam bentuk vector ataupun raster yang mewakili geometri topologi, ukuran, bentuk, posisi dan arah.

b. Data atribut/data tematik

Fungsi pengguna berguna untuk memilih informasi yang diperlukan, membuat standar, update data yang efisien, analisa output untuk hasil yang diinginkan serta merencanakan aplikasi.

II.3.3. Subsistem Utama GIS

GIS terdiri dari empat subsistem utama :

- a. Data input: subsistem ini bertugas untuk mengumpulkan mempersiapkan, dan menyimpan data spasial dan atributnya dalam berbagai sumber. Sub sistem ini pula yang bertanggung jawab dalam mengonversikan atau mentransformasikan format-format data aslinya kedalam format (native) yang dapat digunakan oleh perangkat SIG yang bersangkutan.
- b. Data Output: sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengekspornya ke format yang dikehendaki) seluruh atau sebagian basis data (spasial) baik dalam bentuk *softcopy* maupun *hardcopy* seperti halnya tabel, grafik, *report*, peta, dan lain sebagainya.
- c. Data Management: sub-sistem ini mengorganisasikan baik dalam spasial maupun tabel-tabel atribut terkait kedalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil kembali atau di *retrieve* (*di-load* ke memori), *di-update*, dan *di-edit*.
- d. Data Manipulation & Analysis: sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, sub-sistem ini juga melakukan manipulasi (evaluasi dan penggunaan fungsi-fungsi dan operator matematis & logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan.

II.3.4. Pengetahuan Peta

Menurut Nur Meita Indah Mufidah (2008 ;1), Peta merupakan suatu representasi konvensional (kriteria) dari priter-unsur (fatures) fisik (alamiah dan buatan manusia) dari sebagian atau keseluruhan permukaan bumi di atas media bidang datar dengan skala tertentu.

Adapun persyaratan-persyaratan kriteria yang harus dipenuhi oleh suatu peta sehingga menjadi peta yang ideal adalah:

- a. Jarak antara titik-titik yang terletak di atas peta harus sesuai dengan jarak aslinya di permukaan bumi (dengan memperhatikan priter skala tertentu).
- b. Luas suatu priter yang direpresentasikan di atas peta harus sesuai dengan luas sebenarnya (juga dengan mempertimbangkan skalanya).
- c. Sudut atau arah suatu garis yang direpresentasikan di atas peta harus sesuai dengan arah yang sebenarnya (seperti di permukaan bumi).
- d. Bentuk suatu priter yang direpresentasikan di atas peta harus sesuai dengan bentuk yang sebenarnya (juga dengan mempertimbangkan riter skalanya).

Pada kenyataannya di lapangan merupakan hal yang tidak mungkin menggambarkan sebuah peta yang dapat memenuhi semua kriteria di atas, karena permukaan bumi itu sebenarnya melengkung. Sehingga pada saat melakukan proyeksi dari bentuk permukaan bumi yang melengkung tersebut ke dalam bidang datar (kertas) akan terjadi distorsi. Oleh karena itu maka akan ada kriteria yang tidak terpenuhi, prioritas kriteria dalam melakukan proyeksi peta tergantung dari penggunaan peta tersebut di lapangan misalnya untuk peta yang digunakan untuk perencanaan Jaringan Telekomunikasi maka yang akan jadi prioritas peta ideal

adalah Kriteria 1, sedangkan peta denah kampus yang akan kita digitasi tentunya kriteria 4 yang akan kita utamakan.

II.4. ArcView

Menurut Eko Budiyanto (2010:177), Arcview adalah software yang dikeluarkan oleh ESRI (*Environmental System Research Institute*). Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada Arcview untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan ArcView banyak diterapkan dalam berbagai pekerjaan , seperti analisis pemasaran, perancangan wilayah dan tata ruang, sistem informasi persil, pengendalian dampak lingkungan, bahkan untuk keperluan militer (Eko Budiyanto, 2003:3).

II.5. Pengertian Basis Data (*Database*)

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. database sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. misalnya dari data nama siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut

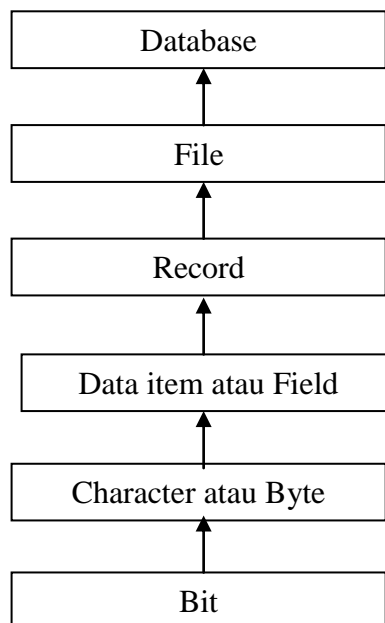
akan anda dapatkan dari software pemroses database dengan cara anda memberikan perintah dalam bahasa tertentu yaitu SQL(*Structured Query Language*). (Wahana Komputer, 2010 : 24)

Dalam database ada sebutan-sebutan untuk satuan data yaitu:

1. Karakter, ini adalah satuan data terkecil. data terdiri atau susunan karakter yang pada akhirnya memawakili data yang memiliki arti dari sebuah fakta.
2. Field, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan database, feld juga disebut atribut.
3. Record, adalah kumpulan dari field. Pada record anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan field-field yang ada.
4. Tabel, adalah sekumpulan dari record-record yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah database.
5. File, adalah bentuk fisik dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi. (Wahana Komputer, 2010 : 24)

II.5.1. Struktur Data

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis (dasar) dalam menyediakan informasi bagi para pemakai. Dimana dalam membentuk *database* yang terstruktur harus memperhatikan jenjang-jenjang dari data dapat dilihat pada gambar dibawah ini:



Gambar II.3. Hierarki data

Sumber : Janner Simarmata (2010:63)

1. *Bit*

Bit merupakan bagian data yang kecil, dapat berupa karakter numerik, huruf maupun karakter-karakter khusus yang membentuk suatu item data, dimana kumpulan karakter membentuk satu *field*.

2. *Byte*

Byte adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. sistem *biner* merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin, yang merupakan serangkaian komponen elektronik dan hanya dapat membedakan 2 macam keadaan, yaitu ada tegangan dan tidak ada tegangan yang masuk ke rangkaian tersebut.

3. *Field*

Suatu *field* menggambarkan suatu atribut *record* yang menunjukkan suatu item dari data seperti nama, alamat, dimana kumpulan *field* membentuk suatu *record*.

4. *Record*

Suatu *record* menggambarkan satu kesatuan data yang sejenis, dimana kumpulan dari file-file membentuk *database*.

5. *File*

Suatu file menggambarkan kumpulan dari beberapa *record* yang dapat menampung data-data.

6. *Database*

Suatu database menggambarkan data yang saling berhubungan antara satu dengan yang lainnya.

II.5.2. Relasi Data atau ERD (*Entity Relation data*)

Menurut Budi Sutedjo (2010 : 130). *Entity Relation Diagram* berfungsi untuk menggambarkan relasi dari dua file atau dua tabel yang dapat digolongkan menjadi dalam tiga macam bentuk relasi, yaitu satu-satu, satu-banyak, dan banyak ke banyak.

Penggambaran ini akan membantu analisa sistem dalam melakukan perancangan proses yang kelak akan dituangkan dalam bentuk garis-garis program.

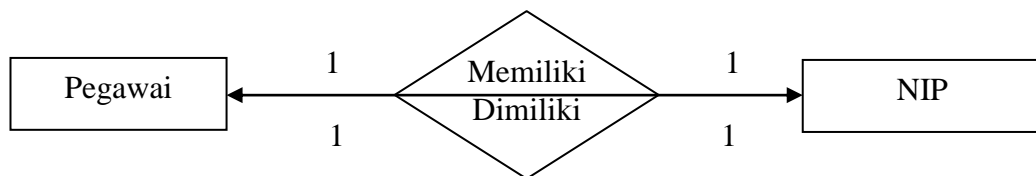
Ir. Harianto Kristanto (Relasi adalah hubungan antara satu *file* atau tabel yang lain dalam suatu database, atau hubungan antara dua atribut dalam suatu *file*.

Relasi antara dua *file* atau tabel dapat di kategorikan menjadi tiga bagian yaitu:

a. Relasi *one to one*

Merupakan hubungan antara *file* pertama dengan *file* kedua atau hubungan antara atribut pertama dengan atribut kedua adalah satu berbanding dua.

Contoh Relasi *one to one* dapat dilihat pada gambar dibawah ini:

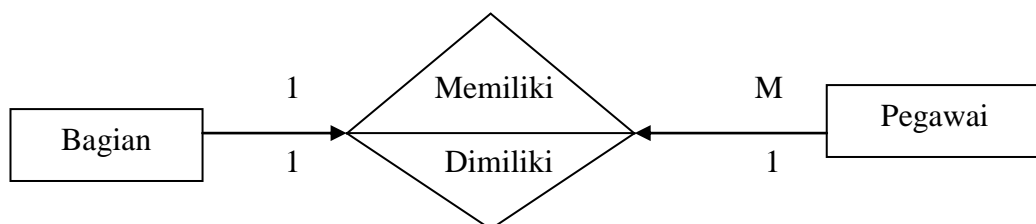


Gambar II.4. Relasi one to one

Sumber : Janner Simarmata (2010:67)

b. Relasi *one to many*

Merupakan hubungan antara *file* pertama dengan *file* kedua atau hubungan antara *atribut* pertama dengan atribut kedua adalah satu berbanding banyak tetapi tidak sebaliknya. Contoh relasi *one to many* dapat dilihat pada gambar dibawah ini.



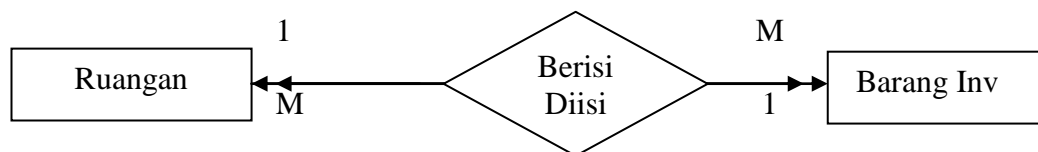
Gambar II.5. Relasi one to many

Sumber : Janner Simarmata (2010:67)

c. *Relasi many to many*

Merupakan hubungan antara *file* pertama dan *file* kedua atau hubungan antara *atribut* pertama dengan atribut kedua adalah banyak berbanding banyak dan sebaliknya.

Contoh relasi *many to many* dapat dilihat pada gambar dibawah ini.




Gambar II.6. Relasi many to many

Sumber : Janner Simarmata (2010:67)

Adapun simbol-simbol pada ERD (*Entity Relation data*), yaitu:

Tabel II.1. Simbol pada ERD

No.	Simbol	Keterangan
1.		<i>Entitas</i> yaitu suatu kumpulan objek atau suatu yang dapat dibedakan atau dapat didefinisikan secara unik
2.		<i>Relationship</i> , untuk melakukan hubungan yang terjadi antara satu atau lebih entitas
3.		<i>Atribut</i> yaitu karakteristik dari entitas atau relationship. Dengan kata lain, atribut adalah kumpulan elemen data yang

		membentuk suatu entitas
4.		<i>Line</i> untuk menghubungkan antar simbol pada ERD

Sumber : Jogiyanto (2012:325)

II.6. Kamus Data

Kamus Data atau *Data Dictionary* atau disebut juga dengan *System Data Dictionary* adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak hingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar cara penulisan). Dengan menggunakan kamus data analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis, kamus digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

Adapun simbol-simbol kamus data yaitu :

Tabel II.2. Simbol – simbol kamus data

No.	Simbol	Keterangan
1.	=	Terdiri dari, mendefinisikan, diuraikan menjadi
2.	+	Dan

3.	()	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field numeric pada struktur file
4.	{ }	Menunjukkan elemen-elemen repetitive, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaankeadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.
5.	[]	Menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada seara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain. (dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)
6.		Pemisah sejumlah alternatif pilihan antara symbol []
7.	@	Identifikasi atribut kunci
8.	**	Komentar

Sumber : Rosa A.S-M.Shalahuddin (2011: 68)

II.7. Teori UML (*Unified Modeling Language*)

Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegoisasikan, dapat dikatakan kita berada pada tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool / model untuk merancang pengembangan software yang berbasis object oriented adalah UML.

Obyek dalam '*software analysis & design*' adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana obyek adalah mobil, manusia, *alarm* dan lain-lainnya. Tapi obyek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database*, *event*, *system messages*. Obyek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciri-ciri ini yang akan membedakan obyek tersebut dari obyek lainnya. Alasan mengapa saat ini pendekatan dalam pengembangan software dengan *object-oriented*, pertama adalah *scalability* dimana obyek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua *dynamic modeling*, adalah dapat dipakai untuk permodelan sistem dinamis dan *real time*.

II.7.1. Sejarah Singkat UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan,

membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah system blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software (<http://www.omg.org>).

Pendekatan analisa & rancangan dengan menggunakan model OO mulai diperkenalkan sekitar pertengahan 1970 hingga akhir 1980 dikarenakan pada saat itu aplikasi software sudah meningkat dan mulai kompleks. Jumlah yang menggunakan metoda OO mulai diuji cobakandan diaplikasikan antara 1989 hingga 1994, seperti halnya oleh Grady Booch dari *Rational Software Co.*, dikenal dengan OOSE (*Object-Oriented Software Engineering*), serta James Rumbaugh dari *General Electric*, dikenal dengan OMT (*Object Modelling Technique*).

Kelemahan saat itu disadari oleh Booch maupun Rumbaugh adalah tidak adanya standar penggunaan model yang berbasis OO, ketika mereka bertemu ditemani rekan lainnya Ivar Jacobson dari Objectory mulai mendiskusikan untuk mengadopsi masing-masing pendekatan metoda OO untuk membuat suatu model bahasa yang uniform / seragam yang disebut UML (*Unified Modeling Language*) dan dapat digunakan oleh seluruh dunia.

Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung Booch untuk membuat sebuah project pendekatan metoda yang uniform/seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan draft metoda UML version 0.8 dan diselesaikan serta di release pada bulan oktober 1995. Bersamaan dengan saat itu, Jacobson bergabung dan

UML tersebut diperkaya ruang lingkupnya dengan metoda OOSE sehingga muncul release version 0.9 pada bulan Juni 1996. Hingga saat ini sejak Juni 1998 UML version 1.3 telah diperkaya dan direspons oleh OMG (Object Management Group).

II.7.2. Pengenalan UML

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam '*MS Word*' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & disain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, arsitektur, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

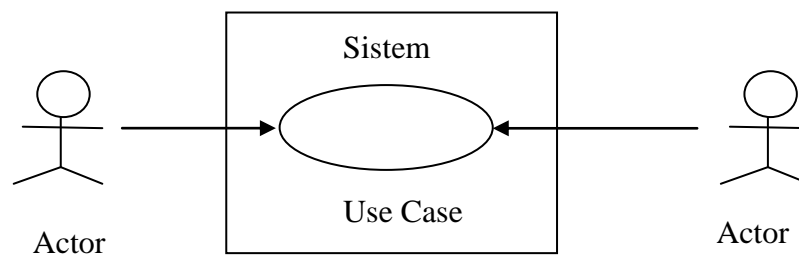
Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologu yang sangat tua. Pada saat masalah ditelusuri dan

spesifikasi dinegosiasikan, dapat dikatakan bahwa kita berada pada tahap rancangan. merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pengembangan software yang berbasis object-oriented adalah UML. Alasan mengapa UML digunakan adalah, pertama, scalability dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan kompleks. Kedua, dynamic modeling, dapat dipakai untuk pemodelan sistem dinamis dan real time. Sebagaimana dalam tulisan pertama, penulis menjelaskan konsep mengenai obyek, OOA&D (Obyek Oriented Analyst/Design) dan pengenalan UML, maka dalam tulisan kedua ini lebih ditekankan pada cara bagaimana UML digunakan dalam merancang sebuah pengembangan software yang disertai gambar atau contoh dari sebuah aplikasi. Adapun jenis-jenis dari tipe diagram UML adalah sebagai berikut :

1. Use Case Diagram

Use Case adalah merupakan pemodelan untuk melakukan (*ehavior*) sistem informasi yang akan dibuat. deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Model *use case* adalah bagian dari *requirement*. Termasuk disini adalah *problem domain object* model dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif *user*. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandanganya. Ide dasarnya bagaimana melibatkan penggunaan

sistem di fase-fase awal analisis dan perancangan sistem. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar sistem. Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan *system/sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Berikut gambar notasi *use case* :



Gambar II.7. Notasi Use Case Diagram

Sumber: Munawar (Pemodelan Visual, 2011:131)

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur sistem dan segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem kelas memiliki apa yang disebut atribut dan metode atau operasi.

Class memiliki tiga area pokok :

- a. Nama (dan stereotype)
- b. Atribut
- c. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.

Protected, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.

- b. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

3. Statechart Diagram




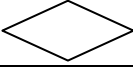


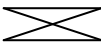
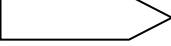
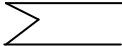
Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).


Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan *dalam* kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

4. Activity Diagram

Activity Diagram adalah menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity Diagram* mempunyai peran seperti halnya lowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa. Berikut adalah simbol yang ada pada activity diagram.

Tabel II.3. Simbol Yang Ada Pada Activity Diagram

SIMBOL	KETERANGAN
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork : digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Rake ; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan

	Aliran Akhir (Flow Final)

Sumber: Munawar (Pemodelan dan UML, 2011:135)

5. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian *langkah-langkah* yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

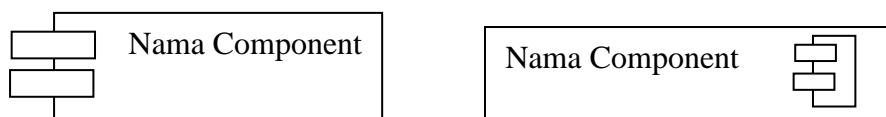
6. Collaboration Diagram

Collaboration diagram adalah perluasan dari *object diagram* (*object diagram* menunjukkan obyek-obyek dan hubungannya satu dengan yang lain). *Collaboration diagram* menunjukkan *message-message* obyek yang dikirimkan satu sama lain. Untuk *menunjukkan* sebuah pesan, buatlah tanda panah di dekat garis asosiasi di antara 2 obyek. Arah panah menunjukkan obyek yang menerima pesan. Label di dekat panah menunjukkan pesannya apa. *Tipikal message* meminta kepada *obyek* yang menerima untuk menjalankan salah satu

operationnya. Sepasang tanda kurung digunakan untuk mengakhiri *message*. Jika ada parameter bisa diletakkan di antara tanda kurung.

7. Component Diagram

Component Diagram dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. *Component Diagram* mengandung *component*, *interface* dan *relationship*. Notasi *component Diagram* dapat dilihat seperti gambar di bawah ini :



Gambar II.8. Notasi Component Diagram

Sumber : Rosa A.S-M.Shalahuddin (2011:125)

Menurut Fowler (2004) hal penting pada *component* adalah *component* mewakili potongan-potongan yang *independent* yang bisa dipesan dan diperbaharui sewaktu-waktu. Dengan demikian pembagian sistem ke dalam *component-component* lebih banyak didorong oleh kepentingan marketing daripada kepentingan teknis.

8. Deployment Diagram

Deployment Diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Deployment Diagram* juga dapat digunakan untuk memodelkan hal-hal berikut:

- a. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
- b. Sistem *client/server*
- c. Sistem terdistribusi murni
- d. Rekayasa ulang aplikasi

II.8. Flow Of Document (FOD)/Flowmap

Flow Of Document (FOD)/Flow Map adalah penggambaran secara grafik dari langkah - langkah dan urutan prosedur dari suatu program. Flowmap berguna untuk membantu analis dan programmer untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif pengoperasian. Biasanya flowmap mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.


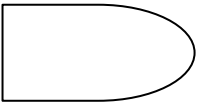
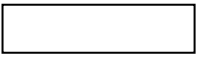
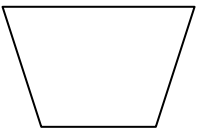
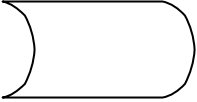

Untuk membuat sebuah analisis menggunakan flowmap seorang analis dan programmer memerlukan beberapa tahapan, diantaranya:

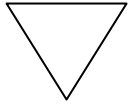

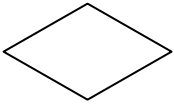
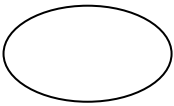
- a. *Flowmap* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
- b. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
- c. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
- d. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
- e. Setiap langkah dari aktivitas harus berada pada urutan yang benar.

- f. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
- g. Gunakan simbol-simbol flowchart yang standar.

Simbol-simbol yang digunakan dalam peta aliran data antara lain:

Tabel II.4. Tabel simbol-simbol Flow Map

No	Simbol	Arti
1		Formulir, Berfungsi sebagai media output dalam bentuk dokumen.
2		Tampilan Ke Monitor
3		Proses ke komputer
4		Proses manual
5		File
6		Disk/Database

7		Arsip
8		Komunikasi Jarak jauh
9		Pilihan
10		Konektor Antar halaman

Sumber: Munawar (2011:135)

II.9. PHP

PHP adalah Bahasa pemrograman script yang paling banyak dipakai saat ini. *PHP* adalah script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh klien. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru atau *up to date*. Semua script *PHP* dieksekusi pada server dimana script tersebut di jalankan.

PHP merupakan perangkat lunak yang gratis (*open source*) dan mempunyai lintas *platform* yaitu dapat digunakan dengan sistem operasi dan *web server* apapun. *PHP* mampu berjalan di Windows dan beberapa versi *Linux*. *PHP* juga dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary*

yang dapat berjalan sebagai *CGI (Common Gate Interface)*, yaitu teknologi untuk menyajikan data yang bersifat dinamis.

PHP dapat mengirimkan *HTTP header*, dapat men set *cookies*, mengauthentication dan re-direct user. *PHP* menawarkan konektisitas dengan baik dengan beberapa basis data antara lain *Oracle, Sybase, MySQL* dan lain – lain.

II.9.1. Keunggulan Pemrograman PHP

Adapun keunggulan pemrograman *PHP* adalah :

1. *PHP* memiliki tingkat akses yang lebih cepat dari pemrograman yang berbasis web lainnya seperti *ASP*.
2. *PHP* memiliki tingkat keamanan yang lebih tinggi karena *server* yang bersangkutan akan memproses semua perintah *PHP* di *server* dan mengirimkan hasilnya ke *web browser*. Dengan demikian pengguna internet tidak dapat melihat kode program yang ditulis dalam *PHP*.
3. *PHP* mampu berjalan di beberapa server yang ada seperti : *Apache, IIS (Internet Information System)* dan lain – lain.
4. *PHP* mampu berjalan di *Linux, Unix dan windows*
5. *PHP* juga mendukung akses ke beberapa database yang susah, antara lain *MySQL, MSOL, dan Windows SQL Server*.

II.10. MySQL Server

Menurut Rulianto Kurniawan (2010:16), MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL termasuk jenis RDBMS (Relational Database Management System). MySQL ini mendukung bahasa pemrograman PHP. MySQL juga mempunyai query atau bahasa SQL (Structured Query Language) yang simple dan menggunakan escape character yang sama dengan PHP. *Software* database mulai bermunculan seiring dengan bertambahnya kebutuhan akan database server. Salah satu dari pendatang baru dalam dunia database ialah *MySQL*, sebuah *server/klien* database *SQL* yang berasal dari Skandinavia. *MySQL* terdiri atas server *SQL*, klien program untuk mengakses *server*, *tools* untuk administrasi, dan *interface* program untuk menulis program sendiri .

Pengembangan *MySQL* dimulai pada tahun 1979 dengan tool database *UNIREG* yang dibuat oleh Michael “Monty” Widenius untuk perusahaan *TcX* di Swedia. Kemudian pada tahun 1994, *TcX* mulai mencari *server SQL* untuk mengembangkan aplikasi Web. Mereka menguji beberapa server komersial namun semuanya masih terlalu lambat untuk table-tabel *TcX* yang besar.

Tahun 1995 David Axmark dari Detro HB berusaha menekan *TcX* untuk me-release *MySQL* di Internet. Dokumentasi *MySQL* yang di-build untuk *GNU configure utility*. *MySQL* 3.11.1 dipublikasikan di dunia tahun 1996 dan didistribusikan untuk Linux dan Solaris. Sekarang ini *MySQL* bekerja untuk banyak platform serta tersedia *source codenya*.

MySQL bukanlah proyek *Open Source* karena lisensi diperlukan juga dalam kondisi tertentu. Akan tetapi *MySQL* dikenal sebagai *software Open Source* karena aturan lisensinya tidak teralu ketat. Selain itu ia juga portable dan bisa dijalankan untuk beberapa system operasi komersial seperti Solaris, *Irix* dan *Windows*.

II.10.1. Kemampuan MySQL Server

Ketika dibandingkan antara *MySQL* dengan *system* database yang lain, maka perlu dipikirkan apa yang paling penting untuk anda. Apakah performa, support, *fitur-fitur SQL*, kondisi keamanan dalam lisensi, atau masalah harga. Dengan pertimbangan tersebut, Anhar (2008:21), *MySQL* memiliki banyak hal yang bisa ditawarkan, antara lain :

a. Kecepatan

Banyak ahli berpendapat *MySQL* merupakan server tercepat.

b. Kemudahan penggunaan

MySQL punya performa tinggi namun merupakan database yang sederhana sehingga mudah disetup dan dikonfigurasi

c. Harga

MySQL cenderung gratis untuk penggunaan tertentu.

d. Mendukung query language

MySQL mengerti bahasa *SQL (Structured Query Language)* yang merupakan pilihan system database modern. Anda juga dapat mengakses *MySQL* lewat *protocol ODBC (Open Database Connectivity)* buatan Microsoft.

e. Kapabilitas

Banyak klien dapat mengakses server dalam satu waktu. Mereka dapat menggunakan banyak database secara simultan.

f. Konetifitas dan sekuritas

Database *MySQL* dapat diakses dari semua tempat di Internet dengan hak akses tertentu.

g. Pertabilitas

MySQL dapat berjalan dalam banyak varian *UNIX* dengan baik, sebaik seperti saat berjalan di *system non-UNIX*.

h. Distribusi yang terbuka

MySQL mudah didapatkan dan memiliki *source code* yang boleh disebarluaskan sehingga bisa dikembangkan lebih lanjut.

Bagaimanapun, mungkin yang paling menarik dari semua karakteristik adalah kenyataan bahwa *MySQL* adalah gratis. Hal ini benar karena *T.c.X* menawarkan *MySQL* sebagai produk gratis untuk umum.

II.10.2. Terminologi Dasar MySQL

Database dalam *MySQL* diklasifikasikan dalam *RDBMS (relational database management system)*. Istilah *RDBMS* ini bisa diartikan sebagai berikut :

- a. “*DB*” yang berarti *database* adalah tempat penyimpana kumpulan informasi yang terdiri atas struktur sebagai berikut :
 1. Koleksi data dalam database dikelompokkan dalam tabel – tabel.
 2. Masing-masig tabel terdiri atas kolom dan baris.
 3. Masing-masing baris memiliki *record*.

4. *Record* dapat berisi informasi yang masing-masing bersesuaian terhadap kolom di atasnya.
- b. “MS” atau *management system* (sistem manajemen) adalah *software* yang membantu anda untuk memasukkan, mengambil, modifikasi, ataupun menghapus *record* dalam *database*.
- c. “R” atau “relational” berarti adanya hubungan yang bagus dalam *DBMS* yaitu terhubungnya informasi yang tersimpan dalam suatu tabel dengan informasi dalam tabel lain.

Untuk berkomunikasi dengan *MySQL* anda dapat menggunakan bahasa yang disebut *SQL (Structured Query Language)*. Saat ini *SQL* adalah bahasa database yang standard dan mayoritas system database mengerti bahasa tersebut. *SQL* memiliki banyak macam perintah dan statement yang akan mendukung system database yang digunakan *MySQL*.

II.11. 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. 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 browser 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*). Dengan protokol ini, komunikasi antar web server dengan client-nya dapat saling dimengerti dan lebih mudah.

Seperti telah dijelaskan diatas, format data pada world wide web adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari. Kata *HyperText* mempunyai arti bahwa seorang pengguna internet dengan web browsernya dapat membuka dan membaca dokumen-dokumen yang ada dalam komputernya atau bahkan jauh tempatnya sekalipun. Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (web) itu.

Proses yang dimulai dari permintaan *webclient* (browser), diterima web server, diproses, dan dikembalikan hasil prosesnya oleh web server ke web client lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap-tiap proses. Secara garis besarnya web server hanya memproses semua masukan yang diperolehnya dari web clientnya.