

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Setiap organisasi atau perusahaan memiliki sistem yang menangani serangkaian aktifitas di dalam suatu sistem agar mencapai tujuan yang optimal. Dalam Laporan Tugas Akhir ini penulis membuat rancangan sistem dengan menggunakan *Web*.

Konsep dasar sistem ini menekankan sekumpulan elemen yang terdapat dalam suatu organisasi, perusahaan, aspek dan sudut pandang yang berbeda sesuai dengan keterangan fungsi dalam hal-hal yang berkaitan dengan sistem yang memiliki ciri dan karakteristik tertentu. Adapun uraian dari konsep dasar sistem akan dijelaskan dibawah ini.

1. Sistem

Secara umum suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen yang terorganisir, saling beinteraksi, saling ketergantungan satu sama lain dan terpadu untuk mencapai tujuan tertentu.

Menurut (Hutahaean (2015:2) “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. “

Dari kutipan diatas, dapat disimpulkan bahwa sistem merupakan serangkaian elemen baik secara fisik maupun non fisik yang berkaitan dengan jaringan kerja yang saling berinteraksi, berhubungan, atau ketergantungan antara satu dengan

yang lain bertujuan untuk menangani dan menyelesaikan serangkaian kegiatan yang terjadi secara rutin dan berulang-ulang.

2. Karakteristik Sistem

Pembuat sistem harus memahami ciri-ciri atau karakteristik yang terdapat pada sekumpulan elemen yang ada, sebagai dasar pertimbangan dalam pembuatan sistem. Supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu (Hutahean, 2014:3) :

1. Komponen

Suatu sistem terdiri dari sejumlah komponen- komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber- sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukkan Sistem (*input*)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem computer program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, *system* akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang. Adapun klasifikasi terhadap sistem (Hutahean, 2014:6) diuraikan sebagai berikut :

1. Klasifikasi sistem sebagai :

- a. Sistem abstrak (*abstract system*) Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.
- b. Sistem fisik (*physical system*) Sistem fisik adalah sistem yang ada secara fisik.

2. Sistem diklasifikasikan sebagai :

- a. Sistem alamiyah (*natural system*) Sistem alamiyah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi.
- b. Sistem buatan manusia (*human made system*) Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine system*).

3. Sistem diklasifikasikan sebagai :

a. Sistem tertentu (*deterministic system*) Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan.

b. Sistem tak tentu (*probabilistic system*) Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

4. Sistem diklasifikasikan sebagai :

a. Sistem tertutup (*close system*) Sistem tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*.

b. Sistem terbuka (*open system*) Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan output dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik.

4. Sistem Informasi

Sistem memiliki tujuan untuk menghasilkan informasi yang nantinya informasi tersebut akan digunakan sebagai dasar pertimbangan untuk pencapaian di dalam suatu organisasi atau perusahaan.

Menurut (Hutahean, 2014:15) mengemukakan bahwa: “Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Dari kutipan di atas disimpulkan bahwa sistem informasi merupakan kompilasi yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang saling memiliki kaitan, hubungan atau interaksi untuk melakukan pengolahan data menjadi informasi dalam suatu organisasi sebagai dasar dalam pengambilan keputusan untuk menentukan langkah di masa mendatang.

Komponen-komponen yang membangun sistem informasi ini dikenal dengan istilah blok bangunan (*building block*). Adapaun uraian dari blok bangunan (Hutahean, 2014:15) yaitu :

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah 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 (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan.

Teknologi terdiri dari unsur utama :

- a. Teknisi (*human ware atau brain ware*)
- b. Perangkat lunak (*software*)
- c. Perangkat keras (*hardware*)

5. Blok basis data (*data base block*)

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kegagalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal

yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

5. Absensi

Menurut (Setiawan, 2015) Absensi atau kartu jam hadir adalah dokumen yang mencatat jam hadir setiap pegawai di suatu perusahaan yang dapat berupa daftar hadir biasa atau kartu hadir yang diisi dengan mesin pencatat waktu.

6. Website

Untuk mengakses internet diperlukan suatu program aplikasi *Web Browser*. Program aplikasi ini mempunyai kemampuan untuk menampilkan suatu *Web Page* yang ditulis dalam format HTML. Dua contoh program aplikasi angkatan pertama adalah *Lynx* dan *Mosaic*. *Lynx* adalah *Web Browser* yang berbasis teks. Kekurangannya adalah tidak bisa menampilkan suatu *Web Page* yang ditulis dalam format HTML. *Mosaic* adalah *browser web* mempopulerkan dikreditkan dengan *World Wide Web*. Itu juga sebelumnya klien untuk protokol seperti *FTP*, *Usenet*, dan *Gopher*. Yang bersih, mudah dipahami *user interface*, keandalan, *Windows* pelabuhan dan instalasi sederhana semua berkontribusi untuk membuat aplikasi yang membuka Web untuk masyarakat umum.

Internet berasal dari kata *Interconnection Networking* yang berarti kata jaringan yang saling berhubungan. Didalam mengatur integrasi dan komunikasi jaringan internet dibutuhkan sebuah *protocol* yang disebut TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Secara umum *internet* dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua informasi bisa didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan.

7. Bahasa Pemrograman

Untuk mengembangkan proses perancangan pembuatan *website* penulis menjelaskan beberapa bahasa pemrograman yang dipakai jika membuat *website*. Diantaranya adalah:

1. HTML (*HyperText Markup Language*)

HTML akronim dari *Hypertext Markup Language*. Menurut (Hidayatullah dan Kawistara, 2014:13) “Yaitu bahasa *standard* yang digunakan untuk menampilkan halaman web”.

Mengapa disebut *hypertext*, karena pemakai dapat membaca tampilan *website* tanpa harus baris demi baris, dapat meloncat dari satu baris ke baris yang lain. Disebut *markup language*, karena susunan instruksinya banyak menggunakan tanda-tanda yang disebut *tag*.

Halaman *web* yang biasa anda lihat ketika *browsing* di internet selalu ditulis menggunakan bahasa HTML. Walaupun halaman *web* tersebut dibuat dengan menggunakan bahasa seperti PHP, JSP, ASPX, hasil akhirnya yang ditampilkan oleh browser tetap adalah HTML murni.

2. PHP (*Personal Home Page*)

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang berhasil *open source*. Menurut (Priyanto & Jauhari, 2015:231), “PHP adalah suatu bahasa *scripting* khususnya digunakan untuk *web*

development. Karena sifatnya yang *server side scripting*, maka untuk menjalankan php harus menggunakan *web server*". Kemudian hasil pemrosesan dikirimkan kepada *web browser* klien. Bahasa pemrograman ini dirancang khusus untuk membentuk web dinamis. Artinya, pemrograman PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk satu tampilan berdasarkan permintaan terkini. Misalnya, anda bisa menampilkan isi *database* ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *common line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

3. CSS (*Cascading Style Sheet*)

Menurut (Rohi Abdulloh 2015:2) "CSS singkatan dari *cascading style sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan HTML terlihat lebih rapi dan indah".

CSS mempunyai 2 bagian utama yaitu *selectors* dan deklarasi. Yang dimaksud *selectors* biasanya *element* HTML yang ingin diubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. Properti sendiri adalah atribut *style* yang di ingin diubah, dan setiap properti memiliki nilai.

8. Basis Data

Dalam pembuatan suatu aplikasi, para pembuat aplikasi atau programmer menggunakan basis data yang digunakan untuk pengolahan data atau penataan file-file yang ada dan digunakan kembali sesuai dengan kebutuhan aplikasi tersebut.

1. Pengertian Basis Data

Basis data berguna sebagai tempat pengolahan informasi yang sangat penting dalam upaya menciptakan suatu aplikasi yang terintegrasi. Menurut (Hidayatullah dan Kawistara, 2014:422) “Basis Data adalah satu komponen utama dalam sistem informasi dan tidak fungsionalitas dalam membuat suatu *web*”.

Dapat ditarik kesimpulan bahwa basis data merupakan kumpulan data yang diolah menjadi informasi dan dapat digunakan kembali jika dibutuhkan dalam suatu organisasi atau perusahaan.

2. Aplikasi Basis Data

Aplikasi basis data sering digunakan oleh para pembuat aplikasi sebagai media pengolahan basis data. Aplikasi basis data yang digunakan penulis dalam perancangan basis data yaitu MySQL. :

1. MySQL

Menurut (Sutanta, 2014:37) “*My Structured Query Language (MySQL)* adalah sebuah perangkat lunak sistem manajemen basis data yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi diseluruh”.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (*Structured Query Language*). SQL

adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah.

Sebagai *software* DBMS, MySQL memiliki sejumlah fitur seperti yang di jelaskan di bawah ini:

a. *Multiplatform*

MySQL tersedia pada beberapa *platform* (*Windows, Linux, Unix* dan *Lain-lain*).

b. Andal cepat dan mudah digunakan

MySQL tergolong sebagai *database server* (*server* yang melayani terhadap permintaan *database*) yang dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan. Berbagai *tool* pendukung juga tersedia (walupun dibuat oleh pihak lain). Perlu diketahui MySQL dapat menangani sebuah tabel yang berukuran dalam *terabyte* (1 *terabyte* = 1024 *Gigabyte*). Namun, ukuran sesungguhnya bergantung kepada batasan sistem operasi. Sebagai contoh : pada sistem Solaris 9/10, batasan ukuran file sebesar 16 *terabyte*.

c. Jaminan ke amanan akses

MySQL mendukung keamanan *database* dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia, sedangkan *user* lain tidak boleh.

d. Dukungan MySQL

Seperti tersirat dalam namanya, MySQL mendukung perintah SQL (*Structured Query Language*). Sebagaimana diketahui, SQL merupakan standar dalam pengaksesan *database* relasional. Pengetahuan akan SQL akan memudahkan siapapun untuk menggunakan MySQL.

2. PhpMyAdmin

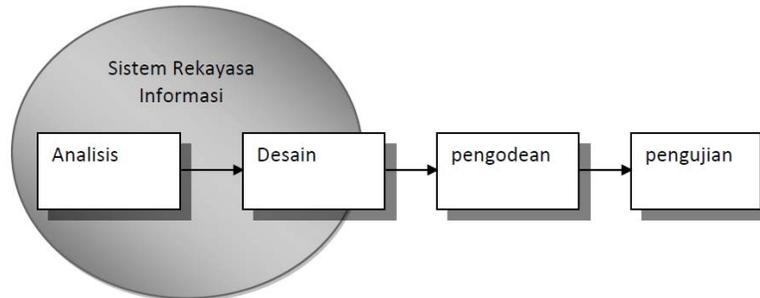
Menurut (Nugroho, 2013) dalam (Abdurahman & Safi, 2018) “phpMyAdmin adalah *tools* yang dapat digunakan dengan mudah untuk manajemen *database* MySQL secara visual dan *server* MySQL, sehingga kita tidak perlu lagi harus menulis *query* SQL setiap akan melakukan perintah operasi *database*”.

Setiap RDBMS (*Relation Database Management System*) seperti *Oracle*, *SQL Server*, *MySQL* dan lain-lain, pasti memiliki *tool* yang dapat digunakan untuk mempermudah pengoperasian *database*. *Oracle* memiliki TOAD. *SQL Server* memiliki *Enterprise Manager* dan *SQL Query Analyzer*. Sedangkan *MySQL* memiliki *tool* atau aplikasi yang disebut *PhpMyAdmin*.

9. Model Pengembangan Perangkat Lunak

Disiplin ilmu yang penulis pilih sebagai dasar untuk menentukan pembuatan aplikasi berdasarkan model pengembangan perangkat lunak yaitu model air terjun (*waterfall*). Menurut Sukanto & Shalahuddin, (2015:28) “Model Waterfall sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model SDLC air terjun (*waterfall*) model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau

terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). Berikut adalah gambar model air terjun:



Sumber: Sukamto dan Shalahuddin, (2015:29)

Gambar II.1

Contoh Ilustrasi Model Waterfall

1. Analisis Kebutuhan Perangkat Lunak Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.
2. Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
5. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*) Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru

a. Teori Pendukung

Berikut ini beberapa teori pendukung yang digunakan dalam membangun *website*, diantaranya adalah:

A. *Entity Relationship Diagram (ERD)* dan *Logical Record Structure (LRS)*

Entity Relationship Diagram (ERD) merupakan sebuah cara untuk menggambarkan sebuah basis data yang menggunakan simbol-simbol beserta hubungan antara simbol-simbol tersebut.

Menurut (Indrajani (2011:109) dalam (Wahid et al., 2016) “*Entity Relationship Diagram (ERD)* adalah sebuah pendekatan *top-bottom* dalam perancangan basis

data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut entitas dan hubungan antara entitas-entitas tersebut digambarkan dalam suatu model atau diagram”.

Dua aturan tersebut yaitu:

1. Setiap *entity* akan diubah ke bentuk sebuah kotak dengan nama *entity* berada di luar kotak dan atribut berada di dalam kotak.
2. Sebuah relasi kadang disatukan dalam sebuah kotak bernama *entity*, kadang dipisah dalam sebuah kotak tersendiri.
3. Aturan pokok di tersebut akan sangat dipengaruhi oleh elemen yang menjadi titik perhatian utama pada langkah transformasi yaitu *cardinality*/kardinalitas.

1. Elemen-Elemen *Diagram*

1. Entitas (*entity*)

Entity menunjukkan obyek-obyek dasar yang terkait didalam sistem obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data.

2. Atribut (*Attribute*)

Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data.

3. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id.

4. Atribut multivalai (*multivalued*)

Field atau kolom data yang butuh disimpati dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5. Relasi (*relation*)

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi diantara dua *entity* yang keterangannya.

6. Asosiasi (*association*)

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity*.

Multiplicity yang dimaksud mengalami perubahan yang mengikuti aturan-aturan sebagai berikut :

1. Setiap entitas akan diubah kebentuk kotak.
2. Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi).
3. Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

B. Unified Modeling Language (UML)

Menurut (Rosa A.S & M. Shalahuddin (2014:137) “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

1. Use Case Diagram

Use case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.



Sumber : <https://www.codepolitan.com>

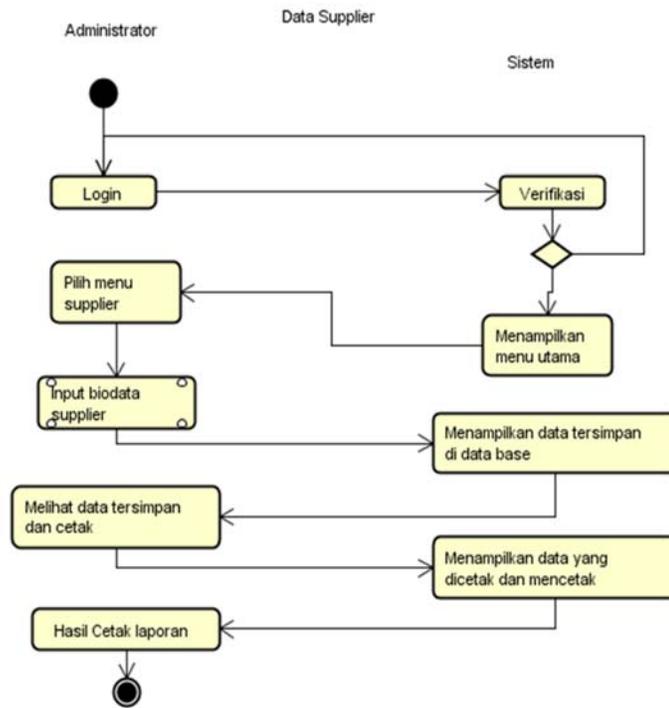
Gambar II.2

Contoh Use Case Diagram

2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu

diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.



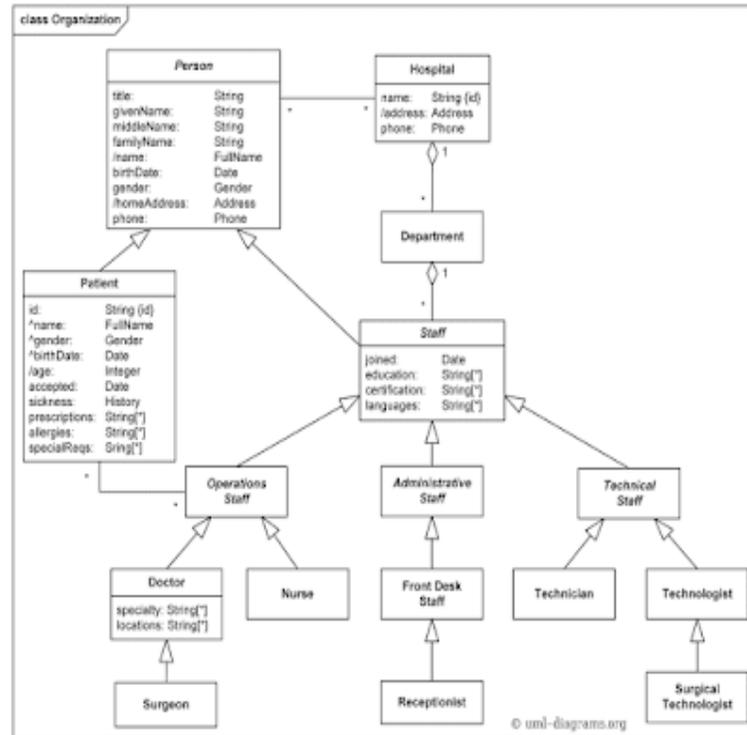
Sumber :<https://pelajarindo.com>

Gambar II.3

Contoh Activity Diagram

3. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.



Sumber : <http://www.sistem-informasi.xyz>

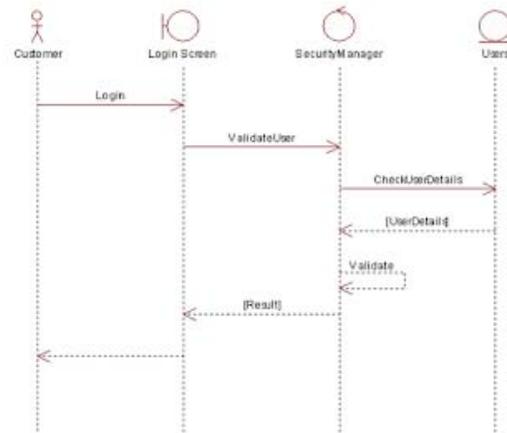
Gambar II.4

Contoh Class Diagram

4. Sequence Diagram

Sequence Diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, sequence diagram juga dapat

menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*.



Sumber :<http://kuliahadsoo.blogspot.com>

Gambar II.5

Contoh Sequence Diagram

C. Struktur Navigasi

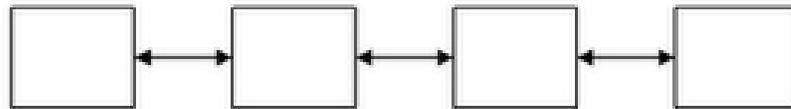
Menurut (Binanto (2010:268) dalam (Anwar et al., 2017) “Struktur Navigasi adalah gabungan dari struktur referensi informasi situs web dan mekanisme link yang mendukung pengunjung untuk melakukan penjelajahan situs”.

Navigasi termasuk struktur terpenting dalam pembuatan suatu aplikasi multimedia dan gambarnya harus sudah ada pada tahap perancangan dan rantai kerja dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan *web* dengan memberikan perintah atau pesan.

Ada 4 macam bentuk dasar dari navigasi yang bisa digunakan dalam proses pembuatan *web*, yaitu:

a. *Linear* (Satu Alur)

Linear (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut. Dengan kata lain struktur ini hanya hanya dapat menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelum atau satu halaman sesudah dan tidak dapat menampilkan dua halaman sebelumnya atau dua halaman sesudahnya. Salah satu yang terpenting dari sturktur ini adalah tidak diperkenankan terjadinya percabangan.

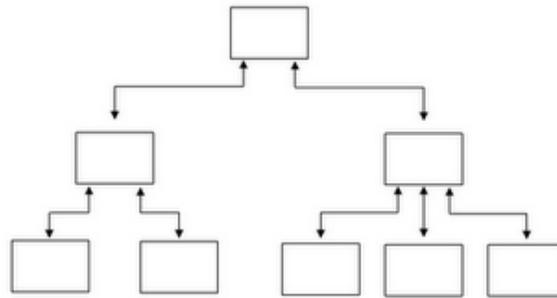


Gambar II.6

Contoh Navigasi *Linear*

b. *Hierarchical* (Hirarki)

Struktur *Hierarchical* ini percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *master page* (halaman utama kesatu), halaman utama ini akan mempunyai halaman percabangan yang dikatakan *slave page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *master page* (halaman utama kedua) dan seterusnya. Yang terpenting dari struktur penjejakan ini tidak diperkenankan adanya tampilan secara *linear*.

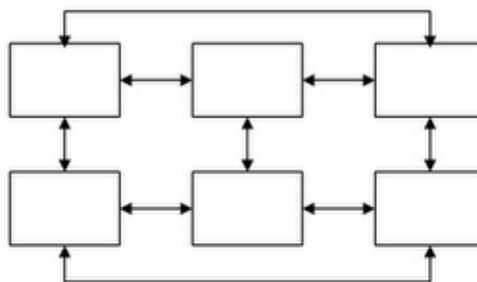


Gambar II.7

Contoh Navigasi *Hierarchical* (Hirarki)

c. *Non Linear* (Tidak Berurut)

Struktur penjejakan *non Linear* (tidak berurut) merupakan pengembangan dari struktur penjejakan *linear*. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute dimana kontrol navigasi dapat mengakses ke semua halaman manapun. Percabangan yang dibuat pada struktur *non linear* ini berbeda dengan percabangan pada *Hierarchical*, karena pada percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master page* dan *Slave page*.

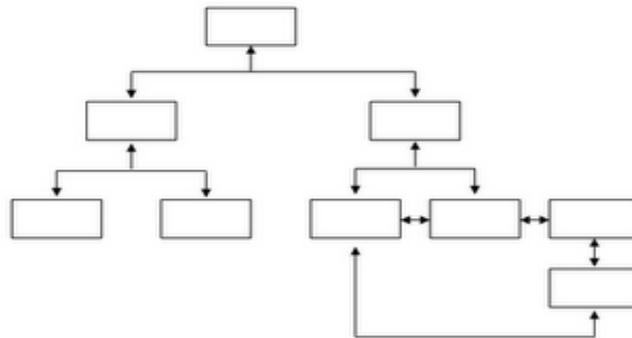


Gambar II.8

Contoh Navigasi *Non Linear*

d. *Composite* (Campuran)

Composite (campuran) atau disebut juga struktur penjejakan bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non Linear*, dan *Hierarchical*. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan, dan bila dalam percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur *Linear* dalam percabangan tersebut. Setiap struktur peta penjejakan seperti yang baru dibahas mempunyai fungsi dan tujuan tersendiri, tidak ada yang lebih baik atau lebih buruk. Pengguna peta penjejakan bergantung kepada kebutuhan dan tujuan dari *Web* yang hendak dibuat. Semakin kompleks peta penjejakan yang digunakan, maka semakin sulit pembuatan *Page* dari peta penjejakan tersebut.



Gambar II.9

Contoh Navigasi *Composite* (Campuran)