

BAB II

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Konsep Dasar Sistem

1. Definisi Sistem

Sebuah sistem yang tepat guna dan sasaran akan memberikan dampak yang positif bagi suatu perusahaan dalam pencapaian sasaran serta tujuan perusahaan. Sistem terdapat beberapa pandangan menurut para ahli, diantaranya:

Menurut Sutarman (2012:13), “Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan yang sama”.

Menurut Shelly-Rosenblatt (2012, p7), “Sistem adalah serangkaian komponen yang saling berhubungan yang menghasilkan hasil tertentu”.

Menurut James A.Hall (2011,p5), “Sistem adalah sebuah kelompok dari dua atau lebih komponen yang saling berhubungan atau subsistem untuk mencapai tujuan bersama”.

Dari definisi diatas,dapat disimpulkan bahwa sistem adalah komponen-komponnen yang saling berhubungan dan berkaitan serta saling bekerjasama untuk mencapai suatu tujuan tertentu.

2.1.2 Konsep Dasar Data dan Informasi

1. Definisi Data

Menurut Considine et al (2012: 7), data adalah fakta mentah yang berhubungan atau menjelaskan suatu kejadian. Sedangkan menurut Rainer dan Cegielski (2011: 10), Data merujuk pada deskripsi dasar suatu benda, kejadian, aktivitas dan transaksi yang direkam, diklasifikasikan, dan disimpan tetapi tidak disusun untuk mengungkapkan arti tertentu. Contohnya angka, huruf, figur, suara, dan gambar.

Dari pengertian diatas dapat disimpulkan bahwa data adalah kumpulan fakta-fakta dan angka mengenai suatu benda, peristiwa, kegiatan dan transaksi yang dicatat, diklasifikasi dan disimpan.

2. Bentuk Data

Menurut Yakub (2012:5) data dapat dibentuk menjadi 5 antara lain:

a. Teks

Teks merupakan sederetan huruf, angka, dan simbol-simbol yang kombinasinya tidak tergantung pada masing-masing item secara individual misalnya, artikel, koran, majalah, dan lain-lain.

b. Data Yang Terformat

Data yang terformat merupakan data dengan suatu format tertentu, misalnya, data yang menyatakan tanggal atau jam, dan nilai mata uang.

c. Citra (*Image*)

Citra (*Image*) merupakan data dalam bentuk gambar, citra dapat berupa grafik, foto, hasil rontsen, dan tanda tangan.

d. Audio

Audio merupakan data dalam bentuk suara misalnya, instrument musik, suara orang, suara binatang, detak jantung, dan lain-lain.

e. Video

Video merupakan data dalam bentuk gambar bergerak dan dilengkapi dengan suara misalnya, suatu kejadian dan aktivitas dalam bentuk film.

3. Sumber Data

Menurut Yakub (2012:6), data dapat diperoleh dari berbagai sumber untuk memperolehnya. Sumber data diklasifikasikan sebagai sumber data internal, sumber data personal, dan sumber data eksternal.

a. Data Internal

Data internal sumbernya adalah orang, produk, layanan, dan proses. Data internal umumnya disimpan dalam basis data perusahaan dan biasanya dapat diakses.

b. Data Personal

Sumber data personal bukan hanya berupa fakta, tetapi dapat juga mencakup konsep, pemikiran dan opini.

c. Data Eksternal

Sumber data eksternal dimulai dari basis data komersial hingga sensor dan satelit. Data ini tersedia di compact disk, flashdisk atau media lainnya dalam bentuk film, suara, gambar, atlas, dan televisi.

4. Definisi Informasi

Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima. Sutarman(2012:14).

5. Kualitas Informasi

Kualitas suatu informasi tergantung dari 3 hal, yaitu informasi harus akurat, tepat waktu, dan relevan. Penjelasan tentang kualitas informasi tersebut dipaparkan di bawah ini menurut Tata sutabri (2012:43)

a. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena biasanya dari sumber informasi sampai penerima informasi ada kemungkinan terjadi gangguan yang dapat mengubah atau merusak informasi tersebut.

b. Tepat Waktu (*Timelines*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usung tidak akan mempunyai nilai lagi karena informasi merupakan suatu

landasan dalam pengambilan keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal bagi organisasi.

c. *Relevance (Relevance)*

Informasi tersebut mempunyai manfaat untuk pemakainya, dimana relevansi informasi untuk tiap-tiap individu berbeda tergantung pada yang menerima dan yang membutuhkan. Nilai informasi ditentukan oleh dua hal yaitu manfaat dan biaya. Suatu informasi dikatakan bernilai apabila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

6. Fungsi Informasi

Menurut Jogiyanto H.M. (2011:10), “Fungsi informasi adalah untuk menambah pengetahuan dan mengurangi ketidakpastian pemakai informasi. Fungsi informasi tidak mengarahkan pengambilan keputusan mengenai apa yang harus dilakukan, tetapi untuk mengurangi keanekaragaman dan ketidakpastian yang menyebabkan diambilnya suatu keputusan yang baik”.

2.1.3 Konsep Dasar Sistem Informasi

1. Konsep Dasar Sistem Informasi

Sistem informasi adalah sistem yang dapat didefinisikan dengan mengumpulkan, memperoses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu. Seperti sistem lainnya, sebuah sistem informasi terdiri atas input (data, instruksi) dan output (laporan, kalkulasi). Sutarman (2012:13)

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa “Sistem informasi adalah sebuah sistem yang terdiri dari pengumpulan, pemasukan, pemrosesan data, penyimpanan, pengolahan, pengendalian dan pelaporan sehingga tercapai sebuah informasi yang mendukung pengambilan keputusan didalam suatu organisasi untuk dapat mencapai sasaran dan tujuannya”.

2. Komponen Sistem Informasi

Tata Sutabri (2012:47) mengemukakan bahwa “Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*Building Block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok bangunan tersebut masing-masing saling berinteraksi satu

dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran". Blok bangunan itu terdiri dari:

- a. Blok Masukan (*Input Block*) *Input* mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.
- b. Blok Model (*Model Block*) Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.
- c. 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.
- d. Blok Teknologi (*Technology Block*) Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Blok teknologi terdiri dari teknisi (*Humanware* atau *Brainware*), perangkat lunak (*Software*) dan perangkat keras (*Hardware*).
- e. Blok Basis Data (*Database Block*) Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).
- f. Blok Kendali (*Controls Block*) Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

3. Tujuan Sistem Informasi

Tujuan dari sistem informasi adalah menghasilkan informasi (*Information*) dari bentuk data yang diolah menjadi bentuk yang berguna bagi para pemakainya. (Jogiyanto H.M., 2011:13) Tujuan sistem informasi terdiri dari Kegunaan (*Usefulness*), Ekonomi (*Economic*), Keandalan (*Realibility*), Pelayanan Langgan (*Customer Service*), Kesederhanaan (*Simplicity*), dan Fleksibilitas (*Fleksibility*).

- a. Kegunaan (*Usefulness*) Sistem harus menghasilkan informasi yang akurat, tepat waktu, dan relevan untuk pengambilan keputusan manajemen dan personil operasi di dalam organisasi.
- b. Ekonomi (*Economic*) Semua bagian komponen sistem termasuk laporan-laporan, pengendalian-pengendalian, mesin-mesin harus menyumbang suatu nilai manfaat setidaknya sebesar biaya yang dibutuhkan.
- c. Keandalan (*Realibility*) Keluaran sistem harus mempunyai tingkatan ketelitian yang tinggi dan sistem itu sendiri harus mampu beroperasi secara efektif bahkan pada waktu komponen manusia tidak hadir atau saat komponen mesin tidak beroperasi secara temporer.
- d. Pelayanan Langgan (*Customer Service*) Sistem harus memberikan pelayanan dengan baik atau ramah kepada para pelanggan. Sehingga sistem tersebut dapat diminati oleh para pelanggannya.
- e. Kesederhanaan (*Simplicity*) Sistem harus cukup sederhana sehingga terstruktur dan operasinya dapat dengan mudah dimengerti dan prosedurnya mudah diikuti.
- f. Fleksibilitas (*Fleksibility*) Sistem harus cukup fleksibel untuk menangani perubahan-perubahan yang terjadi, kepentingannya cukup beralasan dalam kondisi dimana sistem beroperasi atau dalam kebutuhan yang diwajibkan oleh organisasi.

2.1.4 Konsep Dasar Analisa Sistem

1. Pengertian Analisa Sistem

Analisa sistem dapat didefinisikan sebagai penguraian dari sebuah sistem informasi yang utuh ke dalam bagian-bagian tertentu dari suatu informasi dengan maksud untuk melakukan identifikasi dan evaluasi terhadap permasalahan-permasalahan, kesempatan-

kesempatan, hambatan-hambatan, serta kebutuhan-kebutuhan lainnya yang diharapkan sehingga dapat memberikan perbaikan-perbaikan yang diusulkan.

Menurut Yakub (2012:142), Analisa sistem dapat diartikan sebagai suatu proses untuk memahami sistem yang ada, dengan menganalisa jabatan dan uraian tugas (*business users*), proses bisnis (*business proses*), ketentuan atau aturan (*business rule*), masalah dan mencari solusinya (*business problem and business soulution*), dan rencana-rencana perusahaan (*business plan*).

2. Tahap-tahap Analisa Sistem

Tahap analisa sistem merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya. Tahapan ini bisa merupakan tahap yang mudah jika client sangat paham dengan masalah yang dihadapi dalam organisasinya dan tahu betul fungsionalitas dari sistem informasi yang akan dibuat. Tetapi tahap ini bisa menjadi tahap yang paling sulit jika client tidak bisa mengidentifikasi kebutuhannya atau tertutup terhadap pihak luar yang ingin mengetahui detail-detail proses bisnisnya. Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh seorang analis sistem, diantaranya adalah:

1. *Identify*, yaitu proses yang dilakukan untuk mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analysis*, yaitu melakukan analisa terhadap sistem.
4. *Report*, yaitu membuat laporan dari hasil analisis yang telah dilakukan dalam kurun waktu tertentu.

3. Fungsi Analisa Sistem

Adapun fungsi analisa sistem adalah sebagai berikut:

- a. Mengidentifikasi masalah-masalah kebutuhan pemakai (*user*).
- b. Menyatakan secara spesifik sasaran yang harus dicapai untuk memenuhi kebutuhan pemakai.
- c. Memilih alternatif-alternatif metode pemecahan masalah yang paling tepat.
- d. Merencanakan dan menerapkan rancangan sistemnya. Pada tugas atau fungsi terakhir dari analisa sistem menerapkan rencana rancangan sistemnya yang telah disetujui oleh pemakai.

2.2 Teori Khusus

2.2.1 Konsep Dasar Pembuatan Web

1. Definisi HTML

Menurut Sibero (2012:19), “HTML (*Hyper Text Markup Language*) adalah bahasa yang digunakan pada dokumen web sebagai bahasa untuk pertukaran dokumen web”.

Menurut Kustiyahningsih (2011:13), “HTML kependekan *Hyper Text Markup Language*. Dokumen HTML adalah text file murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. File-file HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada di komputer client (user) sehingga isi informasinya dapat di tampilkan secara visual di komputer pengguna”.

Menurut Sutarman (2012:163), “HTML (*Hypertext Markup Language*) adalah bahasa standar yang digunakan untuk pembuatan halaman web atau word wide web, dengan hypertext dan informasi lain yang akan ditampilkan pada halaman web. Dokumen hypertext bisa berisi teks, gambar, dan tipe informasi lain seperti data file, audio, video, dan program executable”.

Berdasarkan pendapat para ahli yang dikemukakan di atas dapat ditarik kesimpulan bahwa HTML adalah bahasa yang sangat tepat dipakai untuk menampilkan informasi pada halaman web, karena HTML menampilkan informasi dalam bentuk hypertext dan juga mendukung sekumpulan perintah yang dapat digunakan untuk mengatur tampilnya informasi tersebut, sesuai dengan namanya, bahasa ini menggunakan tanda (*markup*) untuk menandai perintah-perintahnya.

2. Definisi PHP

Menurut Sibero (2012:49), “PHP (*Personal Home Page*) adalah pemograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”.

Menurut Kustiyahningsih (2011:114), “PHP (atau resminya PHP: *Hypertext Preprosesor*) adalah skrip bersifat server-side yang di tambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari Personal Home Page Tools. Skrip ini akan membuat suatu aplikasi dapat di integrasikan ke dalam HTML sehingga suatu halaman web tidak

lagi bersifat statis, namun menjadi bersifat dinamis. Sifat server side berarti pengerjaan script dilakukan di server, baru kemudian hasilnya dikirimkan ke browser”.

Berdasarkan pendapat para ahli yang dikemukakan di atas dapat ditarik kesimpulan bahwa PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML.

3. Definisi JavaScript

Menurut Sibero (2012:150), “Java script adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan pada web browser”.

Menurut Kustiyahningsih (2011:65), Java script adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi user artinya di sisi browser bukan di sisi server web. Java Script adalah bahasa yang “case sensitive” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh variabel atau fungsi dengan nama TEST berbeda dengan variable dengan nama test dan setiap intruksi diakhiri dengan karakter titik koma (;).

Berdasarkan definisi di atas, maka dapat disimpulkan javascript adalah suatu bahasa pemrograman yang digunakan untuk membuat sebuah web selain bahasa pemrograman PHP.

4. Definisi JQuery Mobile

Menurut Wibisono (2013:3), jQuery Mobile adalah framework berbasis jQuery yang memudahkan kita untuk membuat web app untuk mobile. Selain jQuery mobile sebenarnya banyak framework lain yang dapat digunakan seperti Sencha, jTouch, DHTMLX Touch, Jo dan lainnya.

Camden and Matthews (2012, p1) jQuery Mobile merupakan *User Interface framework* yang menggunakan jQuery sebagai bagian utama untuk pemogramannya. Tidak seperti *framework* lainnya, jQuery Mobile fokus pada HTML dan CSS dengan cara merubahnya menjadi halaman yang *mobile friendly* dan memungkinkan pengguna untuk berinteraksi.

Camden and Matthews (2012, p4) jQuery Mobile hanya fokus dalam *User Interface* sebuah aplikasi, bukan *programming*-nya, hal ini dibuat agar *user* merasakan tampilan yang sama walaupun menggunakan *mobile operating system* yang berbeda. Untuk memperoleh tampilan yang sama pada setiap *mobile operating system* jQuery Mobile membuat CSS sendiri.

jQuery Mobile memiliki berbagai *method* yang disediakan untuk memudahkan para pengembang dalam mengembangkan halaman *mobile*.

Method yang disediakan oleh jQuery Mobile bergantung kepada kehadiran *library* jQuery, sehingga untuk menggunakan jQuery Mobile dibutuhkan jQuery.

Kelebihan jQuery adalah:

- a. Support banyak platform: Webkit (Android, iOS, Opera, Chrome), Firefox mobile, Windows Phone, Blackberry, Bada, Meego.
- b. Berbasis JQuery yang Populer.
- c. Penggunaanya banyak dan forum aktif

1.2.2 Konsep Dasar Database dan Web Server

1. Pengertian Basis Data

Menurut Indrajani (2011: p48), Sebuah kumpulan data yang saling berhubungan secara logis dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh organisasi.

2. Pengertian DBMS

Menurut Connolly dan Begg (2010: p17), Database Management System (DBMS) adalah suatu sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengatur akses ke basis data.

3. Fungsi DBMS

- a. Kontrol atas redundansi atau pengulangan data.
- b. Konsistensi data.
- c. Informasi yang diperoleh dari data yang sama lebih banyak.
- d. Data dapat dibagikan.
- e. Meningkatkan integritas data.
- f. Meningkatkan keamanan data.

- g. Penetapan standarisasi pelaksanaan.
- h. Skala ekonomi.
- i. Kesimbangan dari kebutuhan yang saling bertentangan
- j. Meningkatkan aksesibilitas dan respon data.
- k. Meningkatkan produktivitas.
- l. Meningkatkan Concurrency.
- m. Meningkatkan layanan backup dan recovery

4. Apache dan Sql Server

Menurut Sibero, (2011:374), “Apache adalah perangkat lunak yang digunakan untuk mengakses dan mengatur konfigurasi Apache web server”.

Menurut Raharjo (2011:21), “Sql Server merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak user”.

5. UML

Menurut Prabowo Pudjo Widodo & Herlawati (2011) UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object.

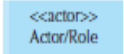
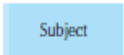
UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

6. Diagram UML

Untuk merancang model sebuah sistem. Ada beberapa diagram yang disediakan dalam UML, antara lain:

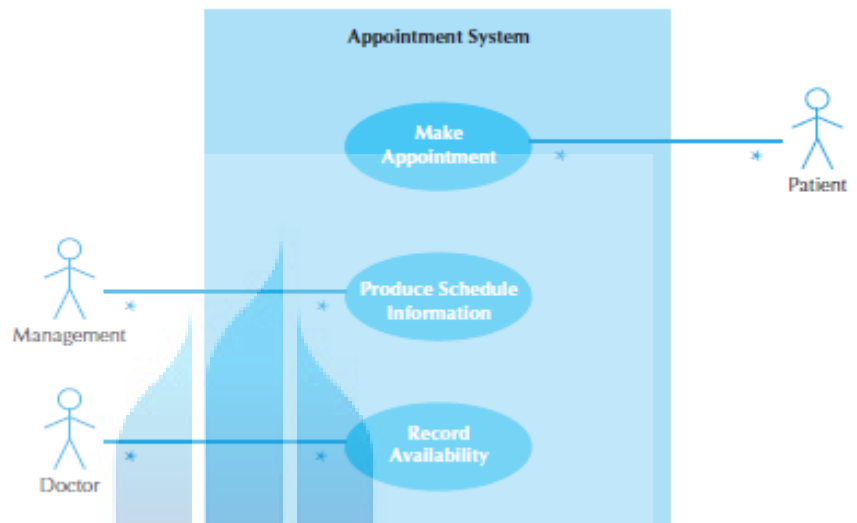
- a. Diagram Use Case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Konsep dasar pemodelan *use case* meliputi :*use case*, aktor, relasi, diagram aktivitas dan diagram *use case*. Aktor merupakan bagian dari *use case* yang bertindak sebagai subjek (pelaku) dalam suatu proses. *Use case* adalah proses-proses yang terjadi dalam suatu *software*. *Use case* juga menggambarkan apa yang sedang dilakukan oleh seorang *actor*. Relasi menggambarkan hubungan antara *actor* dan *use case*.

An actor: - Is a person or system that derives benefit from and is external to the subject. - Is depicted as either a stick figure (default) or if a nonhuman actor is involved, as a rectangle with <<actor>> in it (alternative). - Is labeled with its role. - Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead. - Is placed outside the subject boundary.	 Actor/Role  <<actor>> Actor/Role
A use case: - Represents a major piece of system functionality. - Can extend another use case. - Can include another use case. - Is placed inside the system boundary. - Is labeled with a descriptive verb-noun phrase.	 Use Case
A subject boundary: - Includes the name of the subject inside or on top. - Represents the scope of the subject, e.g., a system or an individual business process.	 Subject
An association relationship: Links an actor with the use case(s) with which it interacts.	
An include relationship: - Represents the inclusion of the functionality of one use case within another. - Has an arrow drawn from the base use case to the used use case.	 <<include>>
An extend relationship: - Represents the extension of the use case to include optional behavior. - Has an arrow drawn from the extension use case to the base use case.	 <<extend>>
A generalization relationship: - Represents a specialized use case to a more generalized one. - Has an arrow drawn from the specialized use case to the base use case.	

Sumber: Alan Dennis,2012

Gambar 2.1 Diagram Simbol Use Case

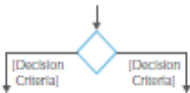
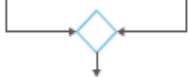


Sumber: Alan Dennis ,2012

Gambar 2.2 Diagram Use Case

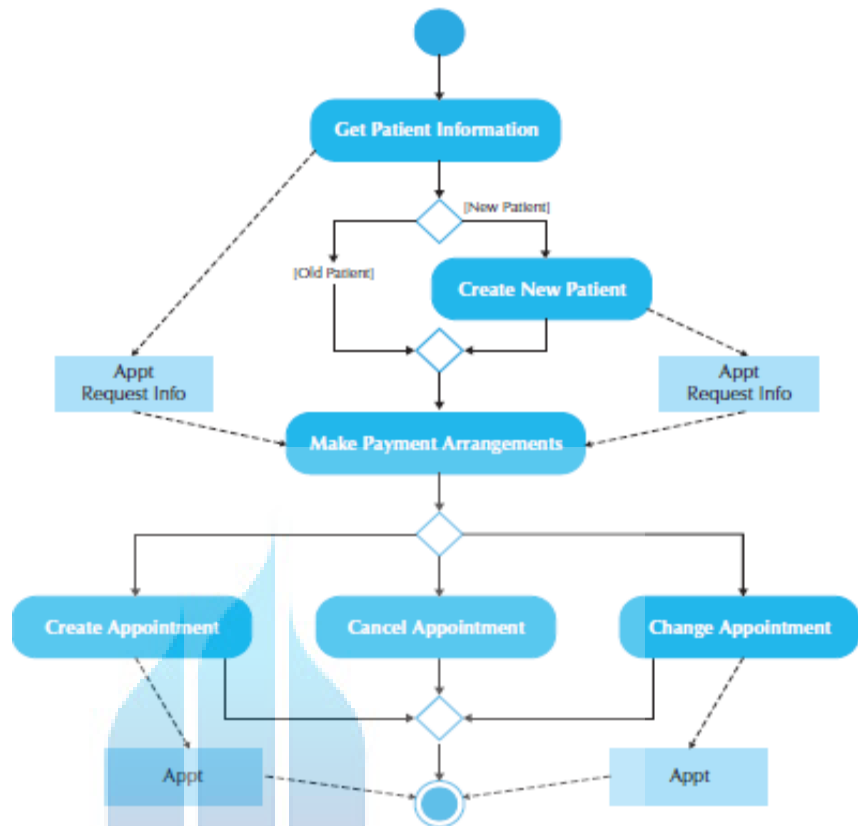
b. Diagram Aktifitas

Diagram aktivitas atau *Activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis. Diagram ini dapat digunakan untuk menggambarkan aliran kerja (*floe of events*) dalam *use case*.

An action: - Is a simple, nondecomposable piece of behavior. - Is labeled by its name.	
An activity: - Is used to represent a set of actions. - Is labeled by its name.	
An object node: - Is used to represent an object that is connected to a set of object flows. - Is labeled by its class name.	
A control flow: Shows the sequence of execution.	
An object flow: Shows the flow of an object from one activity (or action) to another activity (or action).	
An initial node: Portrays the beginning of a set of actions or activities.	
A final-activity node: Is used to stop all control flows and object flows in an activity (or action).	
A final-flow node: Is used to stop a specific control flow or object flow.	
A decision node: - Is used to represent a test condition to ensure that the control flow or object flow only goes down one path. - Is labeled with the decision criteria to continue down the specific path.	
A merge node: Is used to bring back together different decision paths that were created using a decision node.	

Sumber: Alan Dennis, 2012

Gambar 2.3 Diagram Simbol Aktifitas

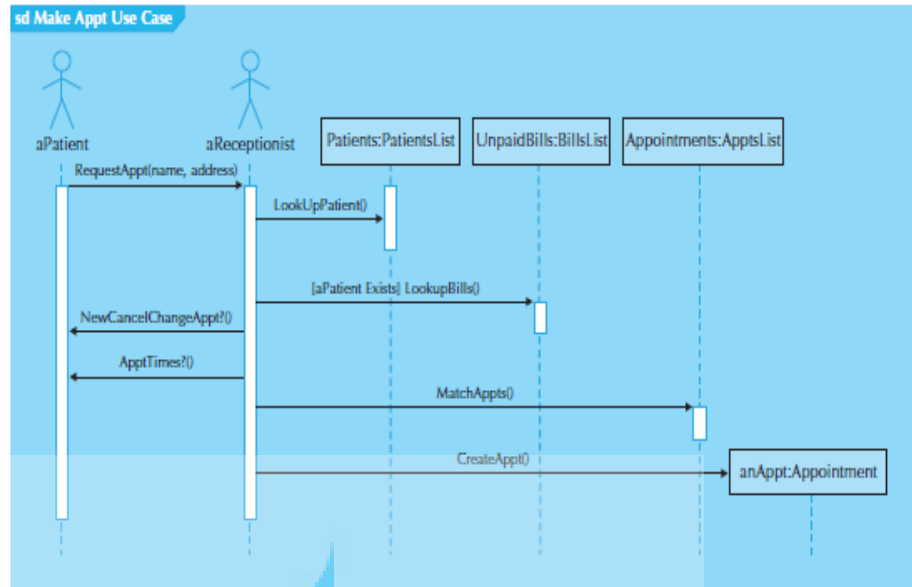


Sumber: Alan Dennis, 2012

Gambar 2.4 Diagram Aktivitas

c. Diagram Sekuensial.

Diagram sekuensial atau *sequence diagram* digunakan untuk menggambarkan aliran fungsionalitas dalam use case. Diagram ini disusun berdasarkan urutan waktu.



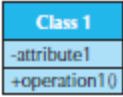
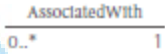
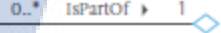
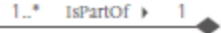
Sumber: Alan Dennis, 2012

Gambar 2.5 Diagram Sekuensial

d. Diagram Kelas

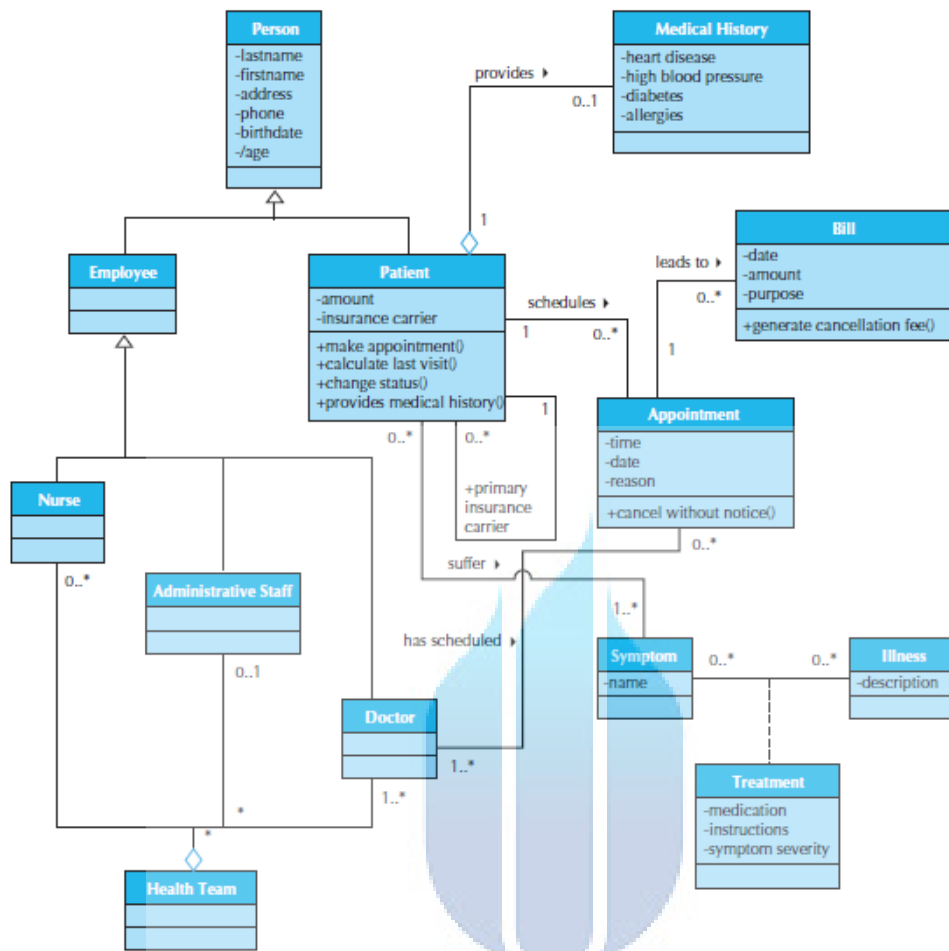
Diagram kelas atau *class diagram* menunjukkan interaksi antar objek dalam sistem. Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Diagram ini menunjukkan gambaran sistem secara statis. Satu diagram kelas menampilkan *subset* dari kelas - kelas dan relasinya.

UNIVERSITAS
MERCU BUANA

A class: - Represents a kind of person, place, or thing about which the system will need to capture and store information. - Has a name typed in bold and centered in its top compartment. - Has a list of attributes in its middle compartment. - Has a list of operations in its bottom compartment. - Does not explicitly show operations that are available to all classes.	 <pre> classDiagram class Class1 { -attribute1 +operation1() } </pre>
An attribute: - Represents properties that describe the state of an object. - Can be derived from other attributes, shown by placing a slash before the attribute's name.	attribute name /derived attribute name
An operation: - Represents the actions or functions that a class can perform. - Can be classified as a constructor, query, or update operation. - Includes parentheses that may contain parameters or information needed to perform the operation.	operation name ()
An association: - Represents a relationship between multiple classes or a class and itself. - Is labeled using a verb phrase or a role name, whichever better represents the relationship. - Can exist between one or more classes. - Contains multiplicity symbols, which represent the minimum and maximum times a class instance can be associated with the related class instance.	 <pre> classDiagram class C1 class C2 C1 "0..*" -- "1" C2 : AssociatedWith </pre>
A generalization: Represents a kind-of relationship between multiple classes.	 <pre> classDiagram class C1 class C2 C1 -- > C2 </pre>
An aggregation: - Represents a logical a-part-of relationship between multiple classes or a class and itself. - Is a special form of an association.	 <pre> classDiagram class C1 class C2 C1 "0..*" o-- "1" C2 : IsPartOf </pre>
A composition: - Represents a physical a-part-of relationship between multiple classes or a class and itself. - Is a special form of an association.	 <pre> classDiagram class C1 class C2 C1 "1..*" *-- "1" C2 : IsPartOf </pre>

Sumber: Alan Dennis, 2012

Gambar 2.6 Diagram Simbol Kelas

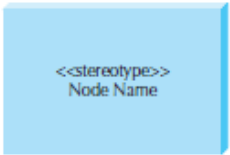
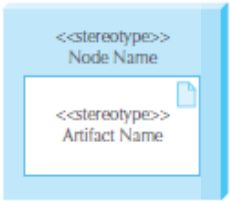
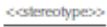


Sumber: Alan Dennis, 2012

Gambar 2.7 Diagram Kelas

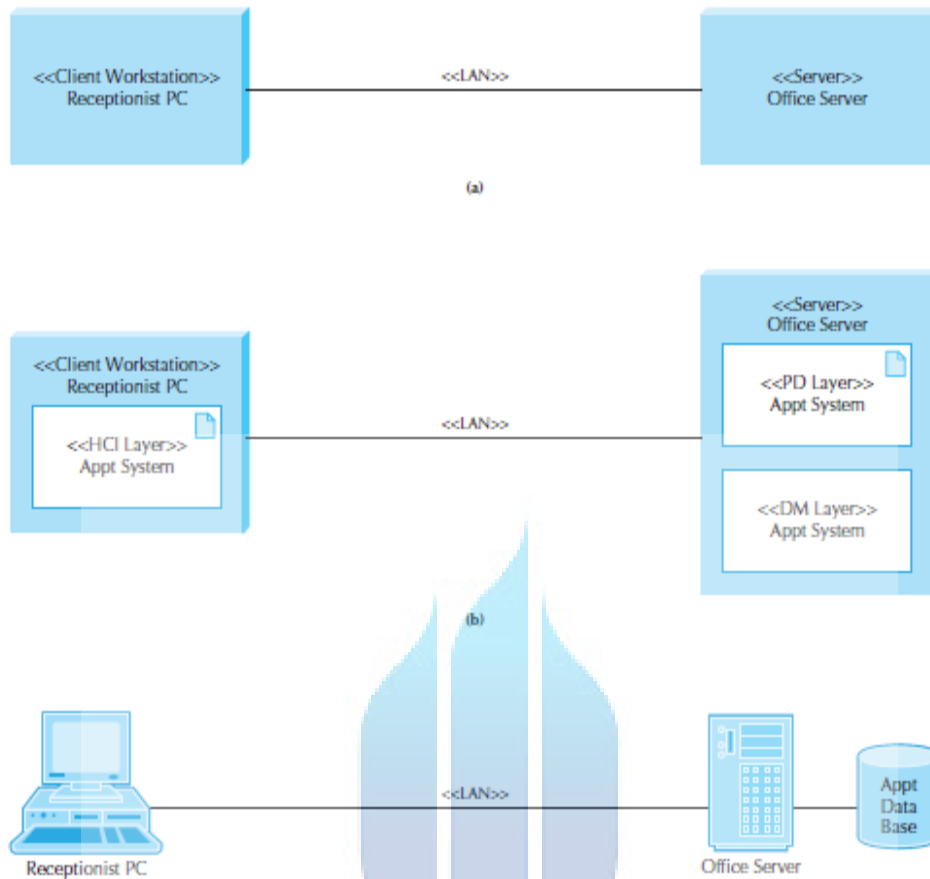
e. Diagram Deployment

Diagram deployment bagian – bagian perangkat lunak suatu sistem ke perangkat keras yang akan mengeksekusinya. Elemen – elemen perangkat lunak seperti komponen, kelas, paket dan sebagainya dimanifestasikan menggunakan artifak serta dipetakan ke perangkat keras yang akan menjalankannya dengan nodes.

A node: ■ Is a computational resource, e.g., a client computer, server, separate network, or individual network device. ■ Is labeled by its name. ■ May contain a stereotype to specifically label the type of node being represented, e.g., device, client workstation, application server, mobile device, etc.	
An artifact: ■ Is a specification of a piece of software or database, e.g., a database or a table or view of a database, a software component or layer. ■ Is labeled by its name. ■ May contain a stereotype to specifically label the type of artifact, e.g., source file, database table, executable file, etc.	
A node with a deployed artifact: ■ Portrays an artifact being placed on a physical node. ■ Supports modeling the distribution of the software over a network.	
A communication path: ■ Represents an association between two nodes. ■ Allows nodes to exchange messages. ■ May contain a stereotype to specifically label the type of communication path being represented, e.g., lan, Internet, serial, parallel, etc.	

Sumber: Alan Dennis 2012

Gambar 2.8 Diagram Simbol Deployment



Sumber: Alan Dennis 2012

Gambar 2.9 Diagram Deployment

1.3 Teori Sales Force Automation

1. Sejarah SFA

Kemunculan sistem SFA dimulai pada tahun 1980-an ketika laptop mulai dikenalkan, kemungkinan dan impian dari salesforce automation mulai dimunculkan oleh perusahaan-perusahaan besar. Tim manajemen mulai memikirkan kemungkinan penggunaan sistem SFA. Pada akhir tahun 1980-an, kisah sukses perihal penggunaan SFA mulai muncul. Beberapa sistem SFA dilaporkan dapat meningkatkan penjualan antara 10 – 30% atau lebih sedangkan investasi yang dikeluarkanpun sudah dapat kembali 100%. Hal ini semakin meningkatkan jumlah perusahaan yang ingin mengimplementasikan sistem SFA.

Sales force automation system merupakan bagian dari salesforce management system yaitu sebuah sistem informasi yang dipergunakan di bagian pemasaran atau manajemen

untuk melakukan proses otomatisasi penjualan atau fungsi manajemen tenaga penjualan. Manfaat dari sales force management system ini adalah untuk meningkatkan performance / kinerja dari salesman / tenaga penjualan, sebagai contohnya:

- a. Salesman tidak perlu menulis laporan penjualan, laporan aktivitas / call sheets, karena semua aktivitas tersebut akan dilakukan di dalam system. Hal ini akan menghemat waktu.
- b. Untuk keperluan pencetakan laporan, dan menyampaikan laporan tersebut kepada sales manager, maka salesman dapat menggunakan sistem intranet untuk mentransmisikan informasi.
- c. Untuk keperluan data inventory, daftar prospek penjualan maupun informasi pendukung penjualan dapat secara mudah di akses, karena semua data telah tersedia. Hal ini akan sangat berguna didalam menentukan rencana pengembangan selanjutnya.

2. Tujuan SFA

Tujuan utama dari suatu sistem SFA adalah dukungan yang diberikan secara otomatis untuk pengumpulan, integrasi, distribusi dan penyajian informasi. Dukungan tersebut bisa mengurangi frustrasi salesforce berkaitan dengan aktifitas- aktifitas non penjualan dengan asumsi informasi yang ada pada sistem tersebut adalah akurat. Selain bermanfaat untuk fungsi penjualan, sales force management system juga berguna bagi para manajemen, manager maupun supervisor di dalam melakukan proses Planning, Organizing, Directing and Controlling terhadap salesman.. Meskipun banyak keuntungan yang didapatkan dengan penggunaan sistem SFA, akan tetapi project sistem SFA berakhir dengan kegagalan, padahal biaya yang sudah dikeluarkan besar.

1.4 Teori Enterprise Resource Planning

Enterprise Resource Planning atau ERP adalah sistem perangkat lunak (software) yang mengintegrasikan manajemen data dan informasi dari keseluruhan fungsional perusahaan yang meliputi keuangan, accounting, produksi, penjualan, pembelian, human resources dan fungsi-fungsi lainnya. Fungsi-fungsi tersebut terpisah oleh modul-modul perangkat lunak, namun saling terhubung dengan satu pusat data yang terintegrasi. Dengan banyaknya fungsional yang terlibat di dalamnya, ERP menjadi sistem yang bersifat *enter once, use many ways*. Artinya, pengguna hanya menggunakan satu akses ke dalam sistem dan akan mendapatkan tampilan serta hak akses sesuai dengan peran dan tanggung jawab (role & responsibilities) yang diberikan oleh perusahaan

1.5 Teori Pengujian

Menurut Nidhra dan Dondeti (2012:1), pengujian software adalah teknik yang sering digunakan untuk verifikasi dan validasi kualitas suatu software. Pengujian software adalah prosedur untuk eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan.

Kesimpulan yang dapat diambil dari pendapat tersebut adalah pengujian software merupakan proses verifikasi dan validasi apakah software memenuhi requirement dan mengidentifikasi kesalahan-kesalahan

yang ditemukan saat eksekusi program.