

BAB II TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

1.1.1 Sistem

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware*, dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain. *Software* mencakup semua perangkat lunak yang dibangun dengan bahasa pemrograman tertentu, pustaka, untuk kemudian menjadi sistem operasi, aplikasi, dan *driver*. Sistem terbentuk oleh subsistem-subsistem yang saling terhubung dan mendukung untuk mencapai sasaran dan menghasilkan sebuah hasil (*output*). Suatu kejadian yang terjadi berulang kali dengan penyelesaian yang serupa dan untuk mencapai sebuah tujuan yang sama dapat diselesaikan oleh suatu sistem yang didesain khusus untuk menangani tujuan tersebut [2].

Sistem adalah serangkaian subsistem yang saling terkait dan tergantung satu sama lain, bekerja bersama-sama untuk mencapai tujuan dan sasaran yang sudah ditetapkannya sebelumnya [3].

Sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur [4].

Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Pada dasarnya sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana, mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu, sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem [5].

Adapun karakteristik yang dimaksud adalah sebagai berikut [5]:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan memengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan serupa suprasistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang memengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara, sedangkan lingkungan luar yang merugikan harus dikendalikan, karena kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian, terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi sistem yang lain.

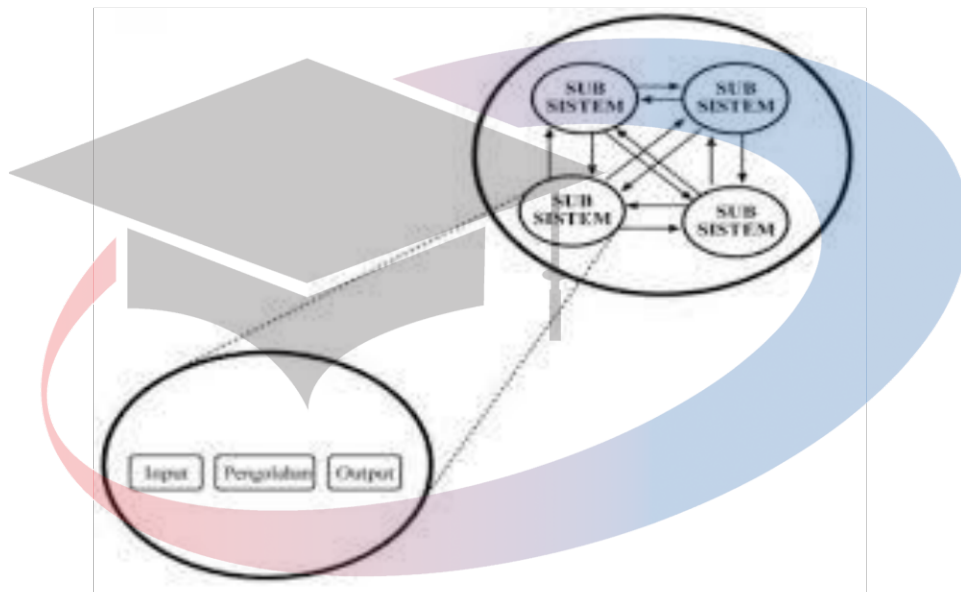
7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

Gambar berikut ini menjelaskan karakteristik dari sebuah sistem [5].



Gambar 2.1 Karakteristik Sistem

1.1.2 Informasi

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat. Proses pengolahan ini memerlukan teknologi. Pada proses pengolahan data, untuk dapat menghasilkan informasi, juga dilakukan proses verifikasi secara akurat, spesifik, dan tepat waktu. Hal ini penting agar informasi dapat memberikan nilai dan pemahaman kepada pengguna. Pengguna dalam hal ini mencakup pembaca, pendengar, penonton, dan bergantung pada bagaimana cara pengguna tersebut menikmati sajian informasi dan melalui media apa informasi tersebut disajikan [2].

Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mempunyai arti bagi si penerima. Sebagai contoh, apabila memasukkan jumlah gaji

dengan jumlah jam bekerja, maka akan mendapatkan informasi yang berguna. Dengan kata lain, informasi datang dari data yang akan diproses [6].

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 (tiga) bagian, yaitu [5]:

1. Informasi Strategis. Informasi ini digunakan untuk mengambil keputusan jangka panjang, mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.
2. Informasi Taktis. Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi Teknis. Informasi ini dibutuhkan untuk keperluan operasional sehari-hari seperti informasi persediaan, retur penjualan, dan laporan kas harian.

Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Informasi yang disampaikan kepada pemakai mungkin merupakan hasil dari data yang dimasukkan ke dalam pengolahan. Akan dapat menambah kemungkinan kepastian atau mengurangi bermacam-macam pilihan [5].

Kualitas dari suatu informasi tergantung dari 3 (tiga) hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timeliness*), dan relevan (*relevance*) [5]:

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi mungkin banyak mengalami gangguan (*noise*) yang dapat mengubah atau merusak informasi tersebut.

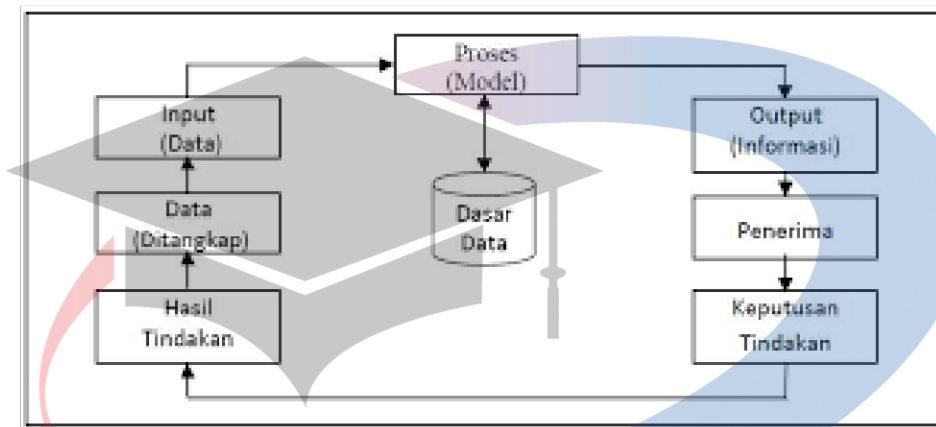
2. Tepat waktu (*timeliness*)

Informasi yang sampai kepada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal bagi organisasi. Dewasa ini informasi bernilai mahal karena harus cepat dikirim dan didapat sehingga memerlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkannya.

3. Relevan (*relevance*)

Infomasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda. Menyampaikan informasi tentang kerusakan mesin produksi kepada akuntan perusahaan tentunya kurang relevan. Akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya, informasi mengenai harga pokok produksi disampaikan untuk ahli teknik merupakan informasi yang kurang relevan, tetapi akan sangat relevan untuk seorang akuntan perusahaan.

Gambar berikut ini menjelaskan pemrosesan data menjadi informasi [5].



Gambar 2.2 Pemrosesan Data Menjadi Informasi

1.1.3 Sistem Informasi

Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan sumber daya manusia yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah data menjadi informasi yang bermanfaat. Di dalamnya juga termasuk proses perencanaan, kontrol, koordinasi, dan pengambilan keputusan. Sebagai sebuah sistem yang mengolah data menjadi informasi yang akan disajikan dan digunakan oleh pengguna, maka sistem informasi merupakan sebuah sistem yang kompleks. Bukan hanya komputer saja yang bekerja (beserta *software* dan *hardware* di dalamnya), namun juga manusia (dengan *brainware* yang dimiliki). Manusia (pengguna/*actor*) dalam hal ini menggunakan seluruh ide, pemikiran, dan perhitungan untuk dituangkan ke dalam sistem informasi yang digunakan [7].

Dalam penerapannya, sebuah sistem informasi dapat berupa sebuah *mainframe*, sebuah *server* dari komputer biasa, maupun *hosting* di internet pada sebuah komputer *server*. Namun tetap saja ada kesamaan di antara ketiga penerapan berbeda ini. Kesamaannya yaitu sama-sama

menggunakan sarana jaringan komputer (internet maupun intranet) untuk melakukan pemrosesan data secara bersama (terdistribusi), baik oleh beberapa pengguna maupun beberapa grup pengguna, menggunakan layanan/fitur/aplikasi yang disertakan [7].

Aktivitas sistem informasi meliputi [7]:

1. Masukan sumber daya data

Data mengenai transaksi bisnis dan kegiatan lainnya harus ditangkap dan disiapkan untuk pemrosesan melalui aktivitas masukan. Masukan biasanya berbentuk aktivitas *entry data* seperti pencatatan dan pengeditan.

2. Pemrosesan data menjadi informasi

Data biasanya tergantung pada aktivitas pemrosesan seperti perhitungan, perbandingan, pemilahan, pengklasifikasian, dan pengikhtisaran. Aktivitas ini mengatur, menganalisis, dan memanipulasi data hingga mengubahnya ke dalam informasi bagi para pemakai akhir.

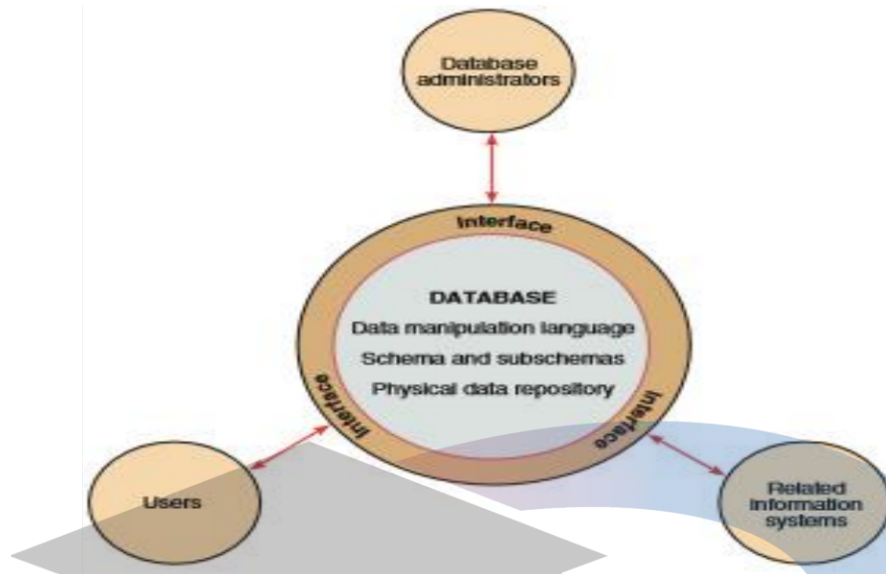
3. Keluaran produk informasi

Informasi dalam berbagai bentuk dikirim ke pemakai akhir dan disediakan untuk aktivitas keluaran. Tujuan dari sistem informasi adalah untuk menghasilkan produk informasi yang tepat bagi para pemakai akhir.

1.2 Basis Data

Basis data atau yang disebut juga *Database Management System* (DBMS) adalah kumpulan alat, fitur, dan antarmuka yang memungkinkan pengguna untuk menambah, memperbarui, mengelola, mengakses, dan menganalisis konten dari *set* data. Dari sudut pandang pengguna, keuntungan utama dari DBMS adalah menawarkan akses data yang tepat waktu, interaktif, dan fleksibel [8].

DBMS menyediakan antarmuka antara *database* dan pengguna yang perlu mengakses data. Meskipun pengguna terutama memperhatikan antarmuka yang mudah digunakan dan dukungan untuk kebutuhan bisnis, seorang analis sistem harus memahami semua komponen DBMS. Selain antarmuka untuk pengguna, administrator basis data, dan sistem terkait, DBMS juga memiliki bahasa manipulasi data, skema dan subskema, serta repositori data fisik. Gambar berikut ini menjelaskan komponen-komponen dari DBMS [8].



Gambar 2.3 Komponen-Komponen DBMS

Ketika pengguna, administrator *database*, dan sistem informasi terkait meminta data dan layanan, DBMS memroses permintaan, memanipulasi data, dan memberikan respon [8]:

1. *Users*

Pengguna biasanya bekerja dengan permintaan yang sudah ditentukan dan perintah *switchboard*, tetapi juga menggunakan bahasa permintaan untuk mengakses data yang disimpan. Bahasa permintaan memungkinkan pengguna untuk menentukan tugas tanpa menentukan bagaimana tugas akan diselesaikan. Beberapa bahasa permintaan menggunakan perintah bahasa alami yang menyerupai kalimat bahasa Inggris biasa. Dengan bahasa *Query By Example* (QBE), pengguna memberikan contoh data yang diminta. Banyak program *database* juga menghasilkan *Structured Query Language* (SQL) yang merupakan bahasa yang memungkinkan *workstation* klien untuk berkomunikasi dengan *server* dan komputer *mainframe*.

2. *Database Administrators*

Database Administrator (DBA) bertanggung jawab atas manajemen dan dukungan DBMS. DBA memperhatikan keamanan dan integritas data, mencegah akses tidak sah, menyediakan cadangan dan pemulihan, jalur audit, memelihara basis data, dan mendukung kebutuhan pengguna. Sebagian besar DBMS menyediakan program utilitas untuk membantu DBA dalam membuat dan memperbarui struktur data, mengumpulkan dan melaporkan pola penggunaan basis data, dan mendeteksi serta melaporkan penyimpangan basis data.

3. *Related Information System*

DBMS dapat mendukung beberapa sistem informasi terkait yang memberikan *input* ke dan membutuhkan data spesifik dari DBMS. Tidak seperti antarmuka pengguna, tidak ada intervensi manusia yang diperlukan untuk komunikasi dua arah antara DBMS dan sistem terkait.

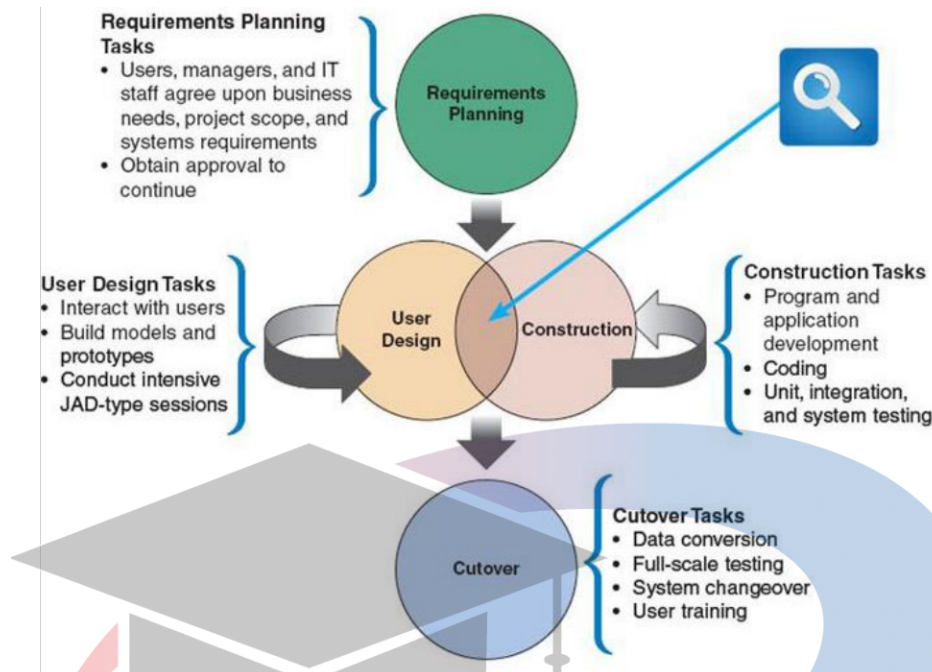
1.3 Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah teknik berbasis tim yang mempercepat pengembangan sistem informasi dan menghasilkan sistem informasi yang berfungsi. Seperti JAD, RAD menggunakan pendekatan kelompok, tetapi melangkah lebih jauh. Produk akhir JAD adalah model persyaratan, sedangkan produk akhir RAD adalah sistem informasi baru. RAD adalah metodologi yang lengkap, dengan siklus hidup empat fase yang sejajar dengan fase SDLC tradisional. Perusahaan menggunakan RAD untuk mengurangi biaya dan waktu pengembangan, dan meningkatkan probabilitas keberhasilan [8].

1.3.1 Fase dan Aktivitas RAD

Model RAD terdiri dari empat fase, yaitu perencanaan kebutuhan, desain pengguna, konstruksi, dan *cutover*. Perhatikan interaksi yang berkelanjutan antara fase desain dan konstruksi pengguna pada gambar berikut ini [8].

UNIVERSITAS
MIKROSKIL



Gambar 2.4 Fase dan Aktivitas RAD

1. *Requirements Planning*

Tahap perencanaan persyaratan menggabungkan elemen-elemen perencanaan sistem dan fase analisis sistem SDLC. Pengguna, manajer, dan anggota staf TI mendiskusikan dan menyetujui kebutuhan bisnis, ruang lingkup proyek, kendala, dan persyaratan sistem. Fase perencanaan persyaratan berakhir ketika tim menyetujui masalah-masalah utama dan mendapatkan izin manajemen untuk melanjutkan.

2. *User Design*

Selama fase desain pengguna, pengguna berinteraksi dengan analis sistem dan mengembangkan model dan prototipe yang mewakili semua proses, *output*, dan *input* sistem. Grup atau subkelompok RAD biasanya menggunakan kombinasi teknik JAD dan alat *case* untuk menerjemahkan kebutuhan pengguna ke dalam model yang berfungsi.

3. *Construction*

Tahap konstruksi berfokus pada tugas pengembangan program dan aplikasi yang mirip dengan SDLC. Namun, dalam RAD, pengguna terus berpartisipasi dan masih dapat menyarankan perubahan atau peningkatan saat layar atau laporan aktual dikembangkan.

4. *Cutover*

Fase peralihan (*cutover*) menyerupai tugas akhir dalam fase implementasi SDLC, termasuk konversi data, pengujian, penggantian ke sistem baru, dan pelatihan pengguna. Dibandingkan dengan metode tradisional, seluruh proses dikompresi.

1.3.2 Objek RAD

Tujuan utama dari semua pendekatan RAD adalah memangkas waktu dan biaya pengembangan dengan melibatkan pengguna dalam setiap fase pengembangan sistem. Karena ini adalah proses yang berkelanjutan, RAD memungkinkan tim pengembangan untuk membuat modifikasi yang diperlukan dengan cepat, seiring dengan perkembangan desain. Di saat anggaran perusahaan sangat ketat, sangat penting untuk membatasi biaya perubahan yang biasanya terjadi dalam jadwal pengembangan yang panjang dan berlarut-larut. Selain keterlibatan pengguna, tim RAD yang sukses harus memiliki sumber daya TI, keterampilan, dan dukungan manajemen. Karena ini adalah proses yang dinamis dan digerakkan oleh pengguna, RAD sangat berharga ketika perusahaan membutuhkan sistem informasi untuk mendukung fungsi bisnis baru. Dengan mendapatkan *input* pengguna dari awal, RAD juga membantu tim pengembangan merancang sistem yang membutuhkan antarmuka pengguna yang sangat interaktif atau kompleks [8].

1.3.3 Keuntungan dan Kerugian Pendekatan RAD

RAD mempunyai keuntungan sebagai berikut [9]:

1. Sangat berguna dilakukan pada kondisi *user* tidak memahami kebutuhan-kebutuhan apa saja yang digunakan pada proses pengembangan perangkat lunak.
2. RAD mengikuti tahapan pengembangan sistem seperti umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada (*reusable object*) sehingga pengembang tidak perlu membuat dari awal lagi dan waktu lebih singkat, berkisar antara 60 hari s.d. 90 hari.
3. Karena mempunyai kemampuan untuk menggunakan komponen yang sudah ada dan waktu yang lebih singkat, maka membuat biaya menjadi lebih rendah dalam menggunakan RAD.

RAD mempunyai kerugian sebagai berikut [9]:

1. Bagi proyek yang besar tetapi berskala, RAD memerlukan sumber daya manusia yang memadai untuk menciptakan jumlah tim RAD yang baik.

2. RAD menuntut pengembang dan pelanggan memiliki komitmen di dalam aktivitas *rapid-fire* yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, maka proyek RAD akan gagal.



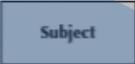




1.4 Use Case Diagram

Sebuah *use case* yang digambarkan berbentuk oval di *Unified Modeling Language* (UML) adalah proses utama yang dilakukan sistem dan yang menguntungkan aktor atau aktor dengan cara tertentu serta dilabeli dengan menggunakan frase kata kerja deskriptif-kata benda [10].

Sebuah *use case* adalah cara formal untuk mewakili cara suatu sistem bisnis berinteraksi dengan lingkungannya. Pada dasarnya, *use case* adalah tinjauan tingkat tinggi dari proses bisnis dalam sistem informasi bisnis. Dari perspektif praktis, kasus penggunaan mewakili seluruh dasar untuk sistem berorientasi objek. Kasus penggunaan dapat mendokumentasikan sistem saat ini (misalnya sistem apa adanya) atau sistem baru yang sedang dikembangkan [10].

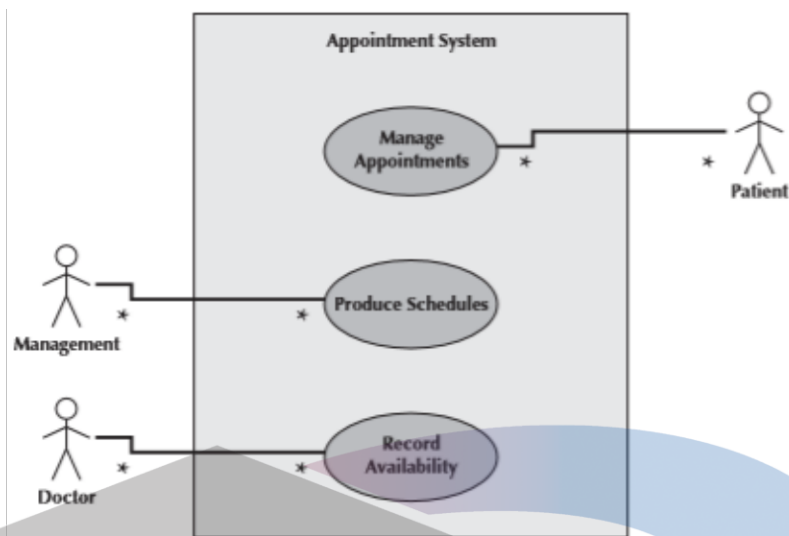
Elemen *Use Case Diagram* termasuk aktor, *use case*, batas subjek, dan serangkaian hubungan antara aktor, aktor dan *use case*, serta *use case*. Hubungan ini terdiri dari asosiasi, termasuk, memperluas, dan hubungan generalisasi. Masing-masing elemen dijelaskan selanjutnya [10].

Gambar tongkat pada diagram mewakili aktor. Seorang aktor bukan pengguna tertentu, tetapi sebaliknya adalah peran yang dapat dimainkan pengguna saat berinteraksi dengan sistem. Seorang aktor juga dapat mewakili sistem lain dimana sistem saat ini berinteraksi. Dalam hal ini, aktor secara opsional dapat diwakili oleh persegi panjang yang berisi aktor. Pada dasarnya, aktor mewakili elemen utama dalam lingkungan dimana sistem beroperasi [10].

<p>An actor:</p> <ul style="list-style-type: none"> ■ 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, 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. 	
<p>A use case:</p> <ul style="list-style-type: none"> ■ 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. 	
<p>A subject boundary:</p> <ul style="list-style-type: none"> ■ 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. 	
<p>An association relationship:</p> <ul style="list-style-type: none"> ■ Links an actor with the use case(s) with which it interacts. 	
<p>An include relationship:</p> <ul style="list-style-type: none"> ■ 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. 	
<p>An extend relationship:</p> <ul style="list-style-type: none"> ■ 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. 	
<p>A generalization relationship:</p> <ul style="list-style-type: none"> ■ Represents a specialized use case to a more generalized one. ■ Has an arrow drawn from the specialized use case to the base use case. 	

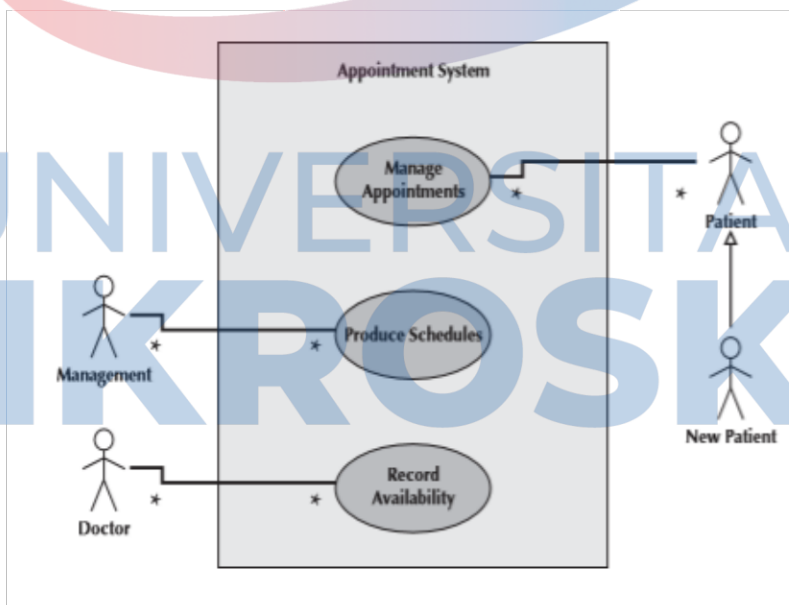
Gambar 2.5 Elemen-Elemen *Use Case Diagram*

Aktor dapat memberikan *input* ke sistem, menerima *output* dari sistem, atau keduanya. Diagram pada gambar berikut ini menunjukkan bahwa tiga aktor akan berinteraksi dengan sistem *appointment* (seorang pasien, seorang dokter, dan manajemen). Terkadang seorang aktor memainkan peran khusus dari tipe aktor yang lebih umum. Sebagai contoh, mungkin ada saat-saat ketika pasien baru berinteraksi dengan sistem dengan cara yang agak berbeda dari pasien umum. Pada kasus ini, aktor khusus (yaitu pasien baru) dapat ditempatkan pada model, ditampilkan menggunakan garis dengan segitiga berlubang di akhir aktor yang lebih umum (yaitu pasien). Aktor khusus mewarisi perilaku aktor yang lebih umum dan meluaskannya dengan cara tertentu [10].



Gambar 2.6 Diagram *Use Case* untuk Sistem *Appointment*

Use case terhubung ke aktor melalui hubungan asosiasi. Hubungan-hubungan ini menunjukkan dengan *use case* dimana para aktor berinteraksi. Garis yang ditarik dari aktor ke *use case* menggambarkan asosiasi. Asosiasi biasanya merupakan komunikasi dua arah antara *use case* dan aktor. Jika komunikasi hanya satu arah, maka panah yang kokoh dapat digunakan untuk menentukan arah aliran informasi [10].



Gambar 2.7 Diagram *Use Case* dengan Aktor Khusus

Misalnya, aktor Pasien berkomunikasi dengan *use case* Kelola Janji Temu. Karena tidak ada panah di asosiasi, komunikasi adalah dua arah. Akhirnya, dimungkinkan untuk merepresentasikan keberagaman asosiasi. Tanda bintang (*) di kedua ujung hubungan antara

Pasien dan *use case* Kelola Janji menunjukkan bahwa seorang pasien individu (contoh dari aktor Pasien) mengeksekusi *use case* Kelola Janji sebanyak yang diinginkan dan bahwa mungkin untuk *use case* Kelola Janji yang akan dieksekusi oleh banyak orang berbeda yaitu beberapa pasien. Dalam kebanyakan kasus, jenis hubungan banyak-ke-banyak ini tepat. Namun, dimungkinkan untuk membatasi jumlah pasien yang dapat dikaitkan dengan *use case Manage Appointments* [10].

1.5 Aplikasi Mobil

Aplikasi mobil adalah sebuah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler, atau *handphone*. Dengan menggunakan aplikasi mobil, maka dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing*, dan lain sebagainya [11].

Beberapa penelitian juga sudah banyak yang menggunakan aplikasi mobil, baik untuk hiburan, mempermudah dalam layanan komunikasi data, maupun sebagai pengendali alat kamera DSLR. Aplikasi mobil dibangun dengan beberapa bahasa pemrograman mobil. Adapun contoh dari *mobile programming* untuk *mobile application* di antaranya adalah *Javafx mobile*, *J2ME*, *C++*, *C# .NET*, dan *Flash Lite* [11].

1.6 Pengembangan Aplikasi

1.6.1 Pengembangan *Front-End*

1.6.1.1 React Native

React Native adalah kerangka kerja JavaScript untuk menulis aplikasi mobil yang asli untuk iOS dan Android. Ini didasarkan pada perpustakaan JavaScript Facebook untuk membangun antarmuka pengguna, tetapi alih-alih menargetkan *browser*, juga menargetkan *platform* seluler. Dengan kata lain, ini memungkinkan pengembang *web* untuk menulis aplikasi seluler yang terlihat dan terasa benar-benar “asli”, semua dari kenyamanan perpustakaan JavaScript. Karena sebagian besar kode yang ditulis dapat dibagikan di antara *platform*, React Native memudahkan pengembangan secara bersamaan untuk Android dan iOS [12].

1.6.1.2 JavaScript

JavaScript diproduksi pada tahun 1995 dan dapat menggunakan program untuk menyimpan halaman di Netscape Navigator *browser*. Bahasa ini telah diadopsi oleh semua penutur

bahasa utama lainnya. JavaScript disediakan sebagai permukaan tradisional untuk menyediakan berbagai bentuk interaktivitas dan kepintaran. *Browser web* bukan satu-satunya *platform* dimana JavaScript digunakan. Beberapa *database*, seperti MongoDB dan CouchDB menggunakan JavaScript sebagai bahasa skrip dan permintaan. Beberapa *platform* untuk pemrograman *desktop* dan *server* terutama proyek Node.js menyediakan lingkungan untuk pemrograman JavaScript di luar *browser* [13].

1.6.2 Pengembangan *Back-End*

Node.js dapat memanfaatkan lingkungan JavaScript di sisi *server*. Penting untuk memahami beberapa konsep inti di balik pilihan desain yang dibuat untuk Node.js dan JavaScript secara umum. Memahami keputusan dan pertukaran akan membuatnya lebih mudah untuk menulis kode yang bagus dan merancang sistem. Ini juga akan membantu menjelaskan kepada orang lain mengapa Node.js berbeda dari sistem lain yang telah digunakan dan dari mana keuntungan kinerja berasal [14].

1.7 *e-Commerce*

Perdagangan elektronik atau *e-commerce* didefinisikan sebagai pembelian dan penjualan produk dan layanan, termasuk dana dan data, secara elektronik. *e-Commerce* dapat dibagi menjadi empat jenis utama, yaitu [15]:

1. *Business to Consumer (B2C)*

Jenis *e-commerce* yang paling umum adalah bisnis ke konsumen (B2C), dimana bisnis menjual produk atau layanan langsung ke konsumen melalui internet. Semua situs ritel pada dasarnya adalah B2C, dengan amazon.com sebagai contoh utama, karena menjadi pengecer *online* terbesar di dunia. Manfaat *e-commerce* mencakup kemampuan bagi pengecer untuk memiliki berbagai macam produk yang tersedia tanpa memerlukan ruang ritel fisik untuk menyimpan dan menampilkan item. Proses pengadaan dan distribusi juga disederhanakan karena produk dikirim langsung ke konsumen dan bukan melalui distributor atau *outlet* ritel. Proses ini memiliki biaya *overhead* yang lebih sedikit dan lebih efisien dalam hal waktu dan uang, mengurangi biaya manajemen persediaan dan mengirimkan produk ke pelanggan dengan biaya lebih rendah daripada *outlet* ritel di dalam toko.

2. *Business to Business (B2B)*

Jenis lain dari *e-commerce* adalah bisnis ke bisnis (B2B), yang menjelaskan transaksi *online* antar bisnis, seperti antara produsen dan grosir, atau antara grosir dan pengecer. Ini juga dapat mencakup layanan yang terkait dengan bisnis, seperti membiarkan ruang komersial, layanan pencetakan, pemasaran *outsourcing*, perekrutan dan penjualan peralatan kantor, dan sebagainya. Seperti B2C, manfaat yang sama dari biaya manajemen persediaan yang lebih rendah dan proses pengadaan dan distribusi yang lebih ramping menjadikan *e-commerce* model yang efisien dan menarik bagi perusahaan yang menjual ke bisnis lain.

3. *Consumer to Consumer (C2C)*

Consumer to Consumer (C2C) e-commerce adalah tempat konsumen menjual produk ke konsumen lain. Secara umum, bisnis C2C menyediakan *platform* bagi berbagai pengguna untuk menggunakannya untuk berinteraksi satu sama lain untuk saling menguntungkan. Contoh *e-commerce* C2C skala besar baru-baru ini adalah Uber. Bisnis hanya menyediakan *platform* transaksional dimana konsumen menawarkan konsumen lain layanan pengangkutan berdasarkan lokasi dan preferensi biaya. eBay adalah bisnis *online* C2C yang diakui secara global. Penjual dapat mendaftarkan produk di eBay dan pembeli yang berminat melelangnya. Situs serupa seperti Airbnb, Gumtree, dan Craigslist, semuanya adalah *platform e-commerce* C2C. Dalam *e-commerce* C2C, konsumen membayar biaya untuk menggunakan *platform* atau *platform* mengambil persentase dari semua transaksi yang diproses melalui situs. Jika *platform* ini gratis bagi semua konsumen untuk membeli dan menjual, maka situs tersebut sering menghasilkan pendapatannya melalui iklan.

4. *Consumer to Business (C2B)*

Consumer to Business (C2B) e-commerce melibatkan konsumen yang menjual produk atau layanan ke bisnis dan bisnis membayar konsumen. Contohnya termasuk Guru.com, situs *web freelancer* yang mempekerjakan, dan situs *web* yang membayar individu untuk menyelesaikan survei *online*.

Mobile commerce (m-commerce) adalah penggunaan perangkat genggam nirkabel seperti ponsel untuk melakukan transaksi komersial *online*. Pertumbuhan *e-commerce desktop* telah melambat dan ponsel terus berkembang, dengan peningkatan tahun ke tahun yang dramatis. Lebih dari sepertiga dari semua transaksi *e-commerce* sekarang terjadi pada perangkat seluler dan ini diperkirakan akan tumbuh sebesar 30% pada tahun 2017 untuk menyalip transaksi *desktop*. Sementara di China, pembeli ponsel sudah mencapai 68% dari pembelian *online*. Pentingnya m-

commerce tidak dapat disangkal dan harus dimasukkan dalam strategi *e-commerce* setiap bisnis [15].

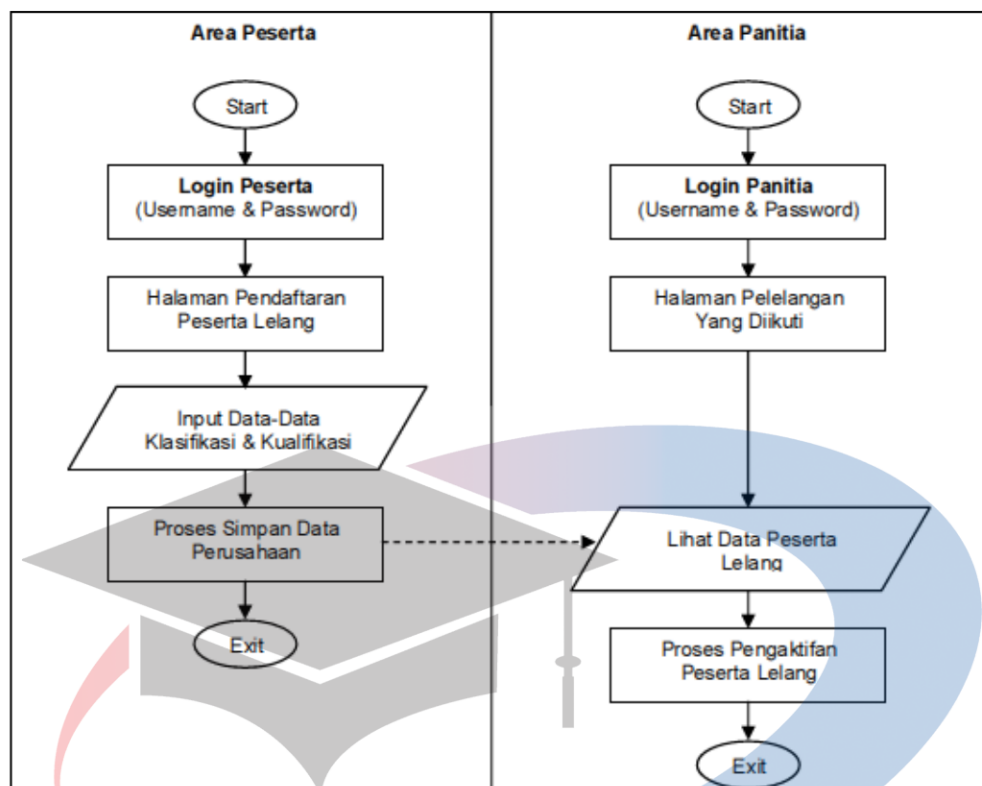
1.8 Lelang

Kata lelang diambil dari kata “*Auctio*”, yang artinya peningkatan secara bertahap. Berbeda dengan jual-beli, lelang merupakan penjualan umum atau penjualan barang-barang yang dilakukan kepada umum dengan harga penawaran yang meningkat atau menurun atau dengan memasukkan harga dalam sampul tertutup, atau kepada orang-orang yang diundang atau sebelumnya diberitahu mengenai pelelangan atau penjualan itu, atau diizinkan untuk ikut serta dan diberi kesempatan untuk menawar harga, menyetujui harga yang ditawarkan, atau memasukkan harga dalam sampul tertutup [16].

Lelang dilaksanakan pada waktu dan tempat tertentu, dan harus didahului dengan pengumuman lelang, serta harus dihadiri oleh Peserta Lelang, Pemohon Lelang, Pemandu Lelang, dan Pejabat Lelang. Lelang di Indonesia harus dilakukan di hadapan Pejabat Lelang dari Kantor Lelang Negara, kecuali ditentukan lain dengan peraturan pemerintah. Dengan demikian, atas pengertian lelang di atas, maka terdapat unsur-unsur yang melekat pada pengertian lelang, yaitu [16]:

1. Penjualan barang (*tender* pengadaan barang dan/atau jasa tidak termasuk dalam pengertian ini)
2. Dilakukan di hadapan umum dengan cara mengumumkannya melalui media massa
3. Pembeli belum diketahui sebelumnya
4. Penawar dengan harga tertinggi akan ditunjuk sebagai pembeli
5. Dilakukan dengan cara penawaran yang khusus
6. Dilakukan pada suatu saat dan tempat tertentu

Gambar berikut ini menggambarkan contoh pengaplikasian lelang [16].



Gambar 2.8 Pengaplikasian Lelang

UNIVERSITAS MIKROSKIL