BAB II
TINJAUAN PUSTAKA

2.1. Pengertian Sistem Informasi

Menurut James O'Brien (2012) sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur. Menurut O'Brien (2012) informasi adalah data yang telah dikonversi ke dalam konteks yang bermakna dan berguna bagi pengguna akhir tertentu. Sehingga dapat ditarik kesimpulan bahwa menurut James O'Brien (2012) sistem informasi dapat merupakan kombinasi teratur dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Manusia bergantung pada sistem informasi untuk melakukan komunikasi dengan peralatan fisik (*hardware)*, instruksi pemrosesan informasi atau prosedur (*software)*, jaringan komunikasi (*network)*, dan data *(data resources)*.

2.2. Komponen-komponen Sistem Informasi

Model sistem informasi menurut James O'Brien (2012) yang menunjukkan kerangka konsep dasar untuk berbagai komponen dan aktivitas sistem informasi. Sistem informasi bergantung pada sumber daya manusia (pemakai akhir dan pakar sistem informasi), *hardware* (mesin dan media), *software* (program dan prosedur), data (dasar data dan pengetahuan), serta jaringan (media komunikasi dan dukungan jaringan) untuk melakukan *input*, pemrosesan, *output*, penyimpanan, dan aktivitas pengendalian yang mengubah sumber daya data menjadi produk informasi.

2.3 *Management Information System* (MIS)

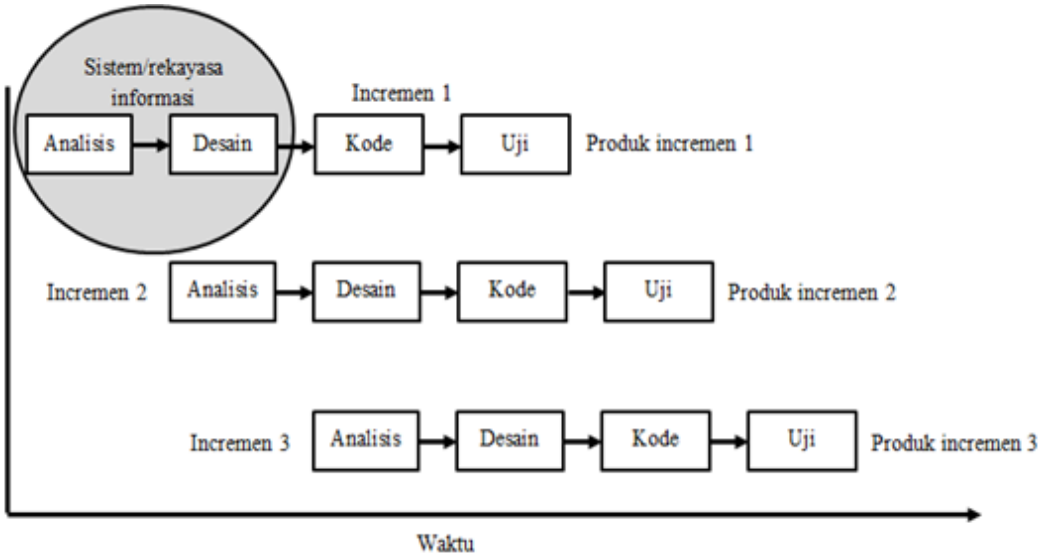
yaitu ketika suatu sistem informasi dirancang untuk menyediakan informasi yang akurat, memiliki rentang waktu dan relevan untuk mendukung efektivitas pengambilan keputusan manajemen (James O'Brien, 2010). Tipe utama dari sistem informasi manajemen memiliki tujuan-tujuan berikut :

1. *Information Reporting Systems* (IRS), menyediakan informasi mengenai produk yang mendukung kebutuhan pengambilan keputusan bagi end user manajerial. Produk informasi berupa tampilan dan laporan yang dapat diatur berdasarkan : permintaan, periode, bergantung pada jadwal yang ditentukan, atau kapanpun diinginkan.
2. *Decision Support Systems* (DSS) merupakan sistem yang interaktif dan berbasis komputer yang menggunakan model keputusan dan spesialisasi basis data untuk membantu proses pengambilan keputusan manajerial. DSS menyediakan model analisis, pembacaan data dan kemampuan presentasi informasi yang memungkinkan para manajer untuk menghasilkan informasi yang mereka butuhkan dalam bentuk yang tidak terstruktur.
3. *Executive Information Systems* (EIS), merupakan sistem informasi manajemen yang dibuat untuk kebutuhan informasi strategis bagi para top manajer. Informasi berasal dari berbagai sumber, termasuk surat-surat, memo, laporan berkala, hasil rapat, telepon dan aktivitas sosial.

2.4 Analisis Model Iteratif

Model iteratif mengkombinasikan proses-proses pada model air terjun dan iteratif pada model prototipe yang menghasilkan versi - versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya. Model ini cocok untuk pengembang dengan turnover staf yang tinggi. Model inkremental dibuat untuk mengatasi kelemahan model waterfall yang tidak mengakomodasi iterasi

dan mengatasi kelemahan dari metode prototipe yang memiliki proses terlalu pendek. (Rosa A.S dan M.Shalahuddin, 2016).



Gambar 2.1 Ilustrasi model iteratif

2.4. Rekayasa perangkat lunak

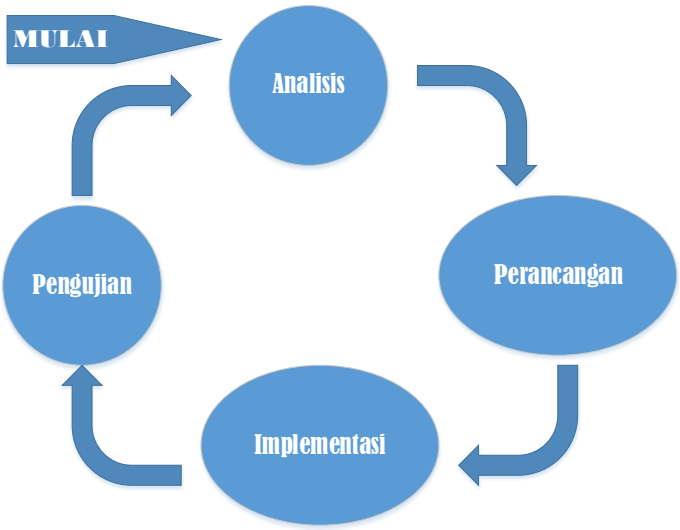
Software engineering adalah pembangunan perangkat lunak dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomis dan mampu bekerja secara efisien menggunakan mesin .Rekayasa perangkat lunak lebih fokus pada pembangunan dan mengirimkan perangkat lunak kepada pelanggan (*customer*). Berikut ini adalah kriteria pembuatan perangkat lunak (Rosa & Shalahuddin, 2015):

1. Perangkat lunak dapat terus dipelihara setelah selesai dibuat seiring berkembangnya teknologi dan lingkungan.
2. Perangkat lunak dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi.
3. Efisien dari segi sumber daya dan penggunaan.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan.

Berdasarkan kriteria diatas maka perangkat lunak yang baik adalah perangkat lunak yang dapat memenuhi kebutuhan pelanggan atau user.

2.5. Proses Rekayasa Perangkat Lunak

Pengembangan teknologi informasi pada saat ini sangat mempermudah dan mempercepat proses rekayasa perangkat lunak dengan adanya bermacam-macam *tools* desain, *tools* pengembangan seperti IDE (*Integrated Development Environment*), *application framework* dan lain-lain. Proses-proses yang dilakukan dalam rekayasa perangkat lunak secara garis besar adalah sebagai berikut:



Gambar 2.3. Tahapan Umum Rekayasa Perangkat Lunak
(Rosa & Shalahuddin, 2015)

Proses perangkat lunak adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari (Rosa & Shalahuddin, 2015):

a. Pengumpulan spesifikasi

   Pengumpulan spesifikasi dilakukan untuk mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

b. Pengembangan

Pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak

c. Validasi

Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan.

d. Evolusi

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan.

2.6. Analisis Sistem

Kegiatan analisis sistem merupakan sebuah kegiatan yang dilakukan untuk melihat sistem yang sudah berjalan, melihat bagian dari sistem yang baik dan tidak baik dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru. Kegiatan analisis juga dapat dilakukan bersamaan dengan kegiatan desain hal ini dilakukan karena pada banyak kasus, *user* sering kesulitan untuk mendefinisikan kebutuhan sistem mereka. Maka dari pada itu *user* akan lebih mudah mendefinisikan kebutuhan jika mereka telah melihat gambaran rancangan sistem yang baru khususnya rancangan antar muka (Rosa & Shalahuddin, 2015):

Sistem analis merupakan seorang spesialis yang mempelajari masalah dan kebutuhan organisasi untuk menentukan bagaimana orang, data, proses, dan teknologi informasi dapat terbaik mencapai perbaikan untuk bisnis. Seorang programmer atau analisis mencakup tanggung jawab dari kedua programmer komputer dan analis sistem. Seorang analis bisnis berfokus hanya pada aspek non-teknis analisis sistem dan desain. Seorang sistem analis harus mampu menyelesaikan berbagai masalah perlu pemecahan, diantaranya (Whitten, Bentley and Dittman, 2004):

1. Masalah, baik yang nyata maupun yang diantisipasi, yang memerlukan tindakan korektif.
2. Kesempatan untuk memperbaiki situasi meskipun tidak adanya keluhan.

3. Arahan untuk mengubah situasi terlepas dari apakah ada orang yang mengeluh tentang situasi saat.

Adapun kemampuan yang dibutuhkan oleh seorang sistem analis

1. Pengetahuan tentang teknologi informasi
2. Komputer pengalaman pemrograman dan keahlian
3. Pengetahuan bisnis umum
4. Umum kemampuan memecahkan masalah
5. Keterampilan komunikasi interpersonal yang baik
6. Keterampilan hubungan interpersonal yang baik
7. Fleksibilitas dan adaptasi
8. Karakter dan etika

2.7. Pengembangan Sistem Informasi

System development process merupakan satu set aktivitas, metode, praktik terbaik, barang siapkirim dan peralatan terotomasi yang digunakan oleh stakeholder untuk mengembangkan secara berkesinambungan serta memperbaiki sistem informasi maupun perangkat lunak. Pendekatan pemecahan masalah secara umum pada pengembangan sistem informasi adalah sebagai berikut (Whitten, Bentley and Dittman, 2004):

1. Identifikasi masalahnya
2. Menganalisis dan memahami masalahnya.
3. Identifikasi persyaratan atau harapan solusi.
4. Identifikasi solusi alternatif dan pilih tindakan "terbaik".
5. Rancang solusi yang dipilih.
6. Terapkan solusi yang dipilih.
7. Evaluasi hasilnya. Jika masalahnya tidak terpecahkan, kembali ke langkah 1 atau 2 yang sesuai.

Proses Pengembangan Sistem yang sederhana terdiri dari:

Tabel 2.1. Proses Pengembangan Sistem

Our Simplified System Development Process	***General Problem-Solving Steps***
System initiation	*1. Identify the problem.*
System analysis	*1. Analyze and understand the problem.* *2. Identify solution requirements or expectations.*
System design	*1. Identify alternative solutions and choose the "best" course of action.* *2. Design the chosen solution.*
System implementation	*1. Implement the chosen solution.* *2. Evaluate the results. If the problem is not solved, return to step 1 or 2 as appropriate.*

1. Inisialisasi sistem - perencanaan awal untuk sebuah proyek untuk menentukan ruang lingkup bisnis awal, sasaran, jadwal, dan anggaran.
2. Analisis sistem - studi tentang domain masalah bisnis untuk merekomendasikan perbaikan dan menentukan persyaratan bisnis dan prioritas untuk solusinya.
3. Perancangan sistem - spesifikasi atau konstruksi solusi teknis berbasis komputer untuk kebutuhan bisnis yang diidentifikasi dalam analisis sistem.
4. Implementasi sistem - konstruksi, instalasi, pengujian, dan pengiriman sistem ke dalam produksi.

Pengembangan sistem informasi memerlukan kerangka untuk mengembangkan sistem tersebut, *Capability Manurity Model* (CMM) merupakan kerangka terstandarisasi untuk menilai tingkat kematangan pengembangan sistem-sistem organisasi sebuah perusahaan baik itu dari proses-proses manajemen maupun dari

proses produksi. Terdapat 5 tingkatan kematangan dalam CMM, berikut adalah tingkatan tersebut (Whitten, Bentley and Dittman, 2004):

1. Tingkat 1 Permulaan

   Pada tingkat ini proyek pengembangan sistem tidak mengikuti proses yang konsisten. Tim pengembang menggunakan peralatan dan metode mereka sendiri. Keberhasilan atau kegagalan biasanya adalah ketrampilan dan pengalaman dari tim tersebut, proses ini tidak dapat ditebak dan tidak dapat diulang.

2. Tingkat 2 Pengulangan

   Proses pengembangan sistem selalu diikuti dan bervariasi dari satu proyek ke proyek lainnya. Praktik-praktik manajemen proyek yang efektif meletakan fondasi untuk proses-proses terstandarisasi pada tingkat berikutnya.

3. Tingkat 3 Terdefinisikan

   Pada tingkat ini, proses pengembangan sistem standar dikembangkan, semua proyek menggunakan versi disesuaikan. Proses ini dilakukan untuk mengembangkan dan merawat sistem informasi pada perangkat lunak.

4. Tingkat 4 Terkelola

   Pada tingkat ini, tujuan-tujuan dapat diukur untuk kualitas dan produktivitas dibentuk. Ukuran-ukuran terinci proses pengembangan sistem informasi standar dan kualitas produk secara rutin berdasarkan data terkumpul. Manajemen lebih pro aktif daripada reaktif terhadap masalah-masalah pengembangan sistem.

5. Tingkat 5 Teroptimisasi

   Pada tingkat ini, proses pengembangan sistem terstandarisasi secara berkesinambungan dan diperbaiki berdasarkan ukuran-ukuran dan analisis data yang dibentuk di tingkat 4.

2.9. Pendekatan Analisis Sistem

Terdapat banyak pendekatan untuk pemecahan masalah, beberapa pendekatan analisis sistem adalah pendekatan terstruktur, teknik informasi, *discovery*

prototiping dan analisis berorientasi objek. Pendekatan pendekatan ini sering dipandang sebagai alternatif-alternatif yang bersaing. Kenyataannya dari berbagai pendekatan tersebut dapat dikombinasikan untuk saling melengkapi satu sama lain. Analisis terstruktur, teknik informasi dan analisis berorientasi objek merupakan contoh-contoh dari *model-driven*.

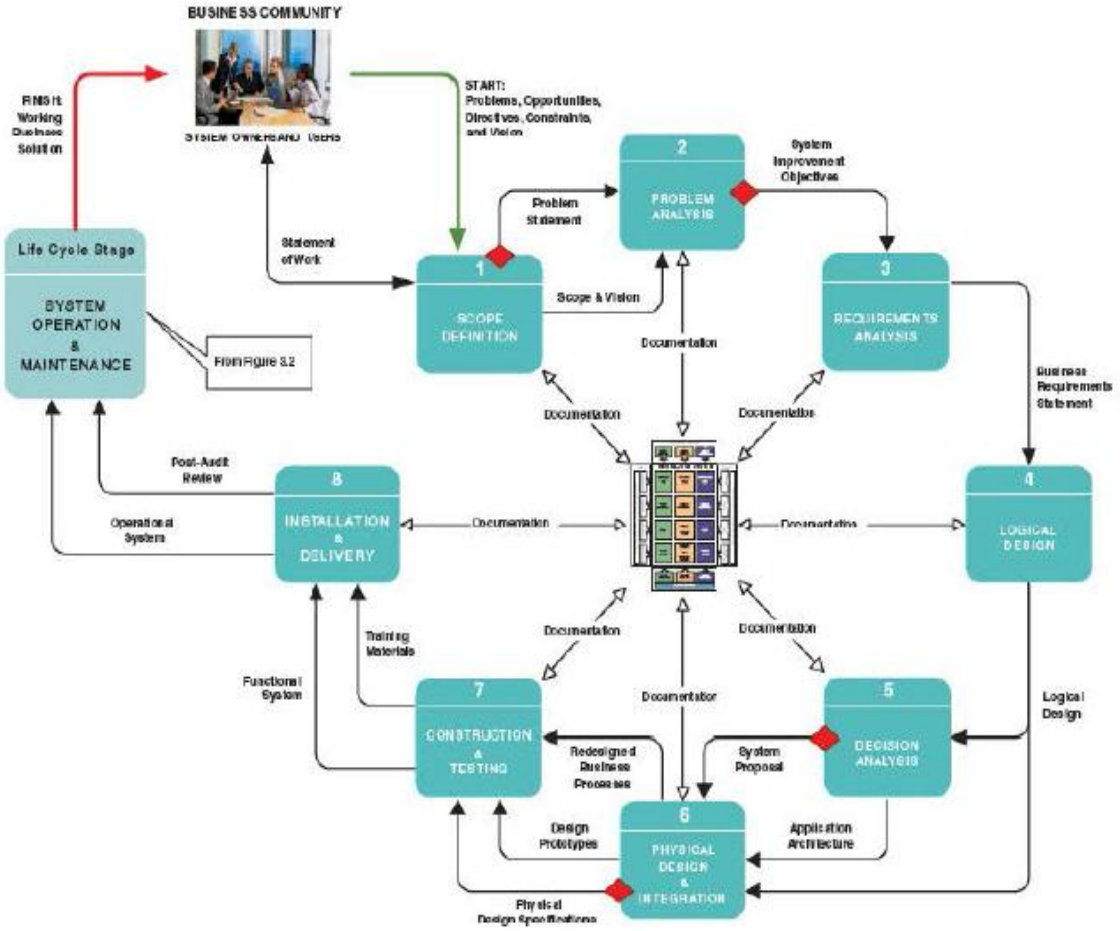
Analisis terstruktur adalah salah satu pendekatan formal pertama untuk analisis sistem informasi dengan berpusat pada proses yang digunakan untuk menganalisis sistem yang ada, mendefinisikan persyaratan-persyaratan bisnis untuk sebuah sistem baru. Teknik informasi adalah suatu pendekatan yang berpusat pada data tapi sensitif pada proses, teknik ini digunakan untuk merencanakan, menganalisis, dan mendesain sistem informasi. Analisis berorientasi objek merupakan sebuah teknik yang mengintegrasikan data dan proses kedalam konstruksi objek atau gambaran-gambaran yang mengilustrasikan objek-objek sistem dari berbagai macam perspektif seperti struktur kelakukan dan interaksi objek (Whitten, Bentley and Dittman, 2004).

2.9.1. Model-Driven

Model-driven analisys merupakan sebuah pendekatan pemecahan masalah yang menekankan pembuatan gambar model-model sistem untuk mendokumentasikan dan memvalidasi sistem yang ada maupun usulan. Analisis dari *model-driven* ini menggunakan gambar untuk mengkomunikasikan masalah-masalah, persyaratan-persyaratan dan solusi bisnis. Contoh model yang mungkin telah dikenal adalah *flowchart*, bagan struktur maupun struktur organisasi (Whitten, Bentley and Dittman, 2004).

Pendekatan-pendekatan *model-driven* hampir selalu diperkuat dengan penggunaan alat yang terotomisasi. Beberapa analis menggambarkan model-model sistem dengan menggunakan perangkat lunak grafis seperti microsoft visio, powerdesigner dan lain. Pendekatan pada *model-driven* dicirikan dalam

metodologi-metodologi dan rute-rute, berikut ini adalah gambaran dari pendekatan analisis *model-driven* dapat dilihat pada gambar 2.4.



Gambar 2.4. Konteks Analisis Sistem
(Whitten, Bentley and Dittman, 2004).

.

1. *Scope Definition* (Definisi Lingkup)

   Fase definisi lingkup adalah fase pertama proses pengembangan sistem klasik, pada pembahasan yang lain fase ini sering disebut dengan fase penyelidikan pendahuluan. Pada tahap ini dilakukan pengumpulan informasi yang akan diteliti tingkat *feasibility* dan ruang lingkup proyek. Hal ini dilakukan untuk menemukan inti dari masalah-masalah yang ada (*problems*), kesempatan untuk meningkatkan kinerja organisasi (*opportunity*) dan kebutuhan-kebutuhan baru yang dibebankan oleh pihak manajemen.

Langkah awal yang harus dikerjakan dalam fase definisi lingkup adalah dengan mengidentifikasi masalah-masalah, kesempatan-kesempatan, perintah dan batasan. Namun hal tersebut harus berdasarkan urgensi, visibilitas, manfaat-manfaat yang nyata dan prioritas. Lingkup juga mendefinisikan batasan-batasan proyek atau aspek-aspek bisnis (Whitten, Bentley and Dittman, 2004).

2. *Problem Analysis* (Analisis Permasalahan)

   Pada tahap ini akan diteliti masalah-masalah yang muncul pada sistem yang ada sebelumnya. Dalam hal ini *project charter* yang dihasilkan dari tahapan *preliminary investigation* adalah kunci utamanya. Hasil dari tahapan ini adalah peningkatan performa sistem yang akan memberikan keuntungan dari segi bisnis perusahaan. Tujuan dari fase analisis masalah adalah mempelajari dan memahami bidang masalah dengan cukup baik untuk secara menyeluruh menganalisis masalah, kesempatan dan batasannya. (Whitten, Bentley and Dittman, 2004).

3. *Requirements Analysis* (Analisis Persyaratan)

   Pada tahap ini akan dilakukan pengurutan prioritas dari kebutuhan-kebutuhan bisnis yang ada. Tujuan dari tahapan ini adalah mengidentifikasi data, proses dan antarmuka yang diinginkan pengguna dari sistem yang baru. Pada titik ini juga mulai terlihat berbagai solusi alternatif, khususnya solusi teknis. Persyaratan pada sistem terbagi menjadi dua yaitu (Whitten, Bentley and Dittman, 2004):

   a. Persyaratan fungsional diidentifikasikan dalam istilah *input, output,* proses dan data tersimpan yang dibutuhkan untuk memenuhi sasaran peningkatan sistem.
   b. Persyaratan nonfungsional merupakan deskripsi mengenai fitur, karakteristik dan batasan lainnya yang menentukan apakah sistem memuaskan atau tidak. Pada persyaratan nonfungsional dapat dilakukan

pengklasifikasian berdasarkan kerangka PIECES (Performa, Informasi, Ekonomi, Kontrol, Efisiensi dan Pelayanan)

4. *Logical Design* (Desain Logis)

   Istilah desain logis harus diinterpresentasikan sebagai teknologi mandiri yang berati makna gambar, menggambarkan sistem independen dari setiap kemungkinan solusi teknis, kebutuhan model bisnis yang diinginkan harus dipenuhi oleh solusi teknis yang ingin dipertimbangkan. Tujuan dari tahapan ini adalah mentransformasikan kebutuhan-kebutuhan bisnis dari fase *requirements analysis* kepada sistem model yang akan dibangun nantinya. Desain logis lebih lanjut mendokumentasikan persyaratan bisnis dengan mengunakan model-model sistem yang menggambarkan struktur data, proses bisnis, aliran data. (Whitten, Bentley and Dittman, 2004).

5. *Decision Analysis* (Analisis Keputusan)

   Fase analisis keputusan adalah untuk mengenali solusi kandidat, menganalisis solusi kandidat dan merekomendasikan sebuah sistem target rancangan. Pada tahap ini akan akan dipertimbangkan beberapa kandidat dari perangkat lunak dan keras yang nantinya akan dipilih dan dipakai dalam implementasi sistem sebagai solusi atas *problems* dan *requirements* yang sudah didefinisikan pada tahapan-tahapan sebelumnya (Whitten, Bentley and Dittman, 2004).

6. *Physical Design* (Desain Fisik)

   Tujuan dari tahapan ini adalah mentransformasikan kebutuhan bisnis yang direpresentasikan sebagai logical design menjadi physical design yang nantinya akan dijadikan sebagai acuan dalam membuat sistem yang akan dikembangkan. Jika di dalam logical design tergantung kepada berbagai solusi teknis, maka *physical design* merepresentasikan solusi teknis yang lebih spesifik (Whitten, Bentley and Dittman, 2004).

7. *Construction and Testing*

   Setelah membuat *physical design*, maka akan dimulai untuk mengkonstruksi dan melakukan tahap uji coba terhadap sistem yang memenuhi kebutuhan-kebutuhan bisnis dan spesifikasi desain. Basis data, program aplikasi, dan antarmuka akan mulai dibangun pada tahap ini. Setelah dilakukan uji coba terhadap keseluruhan sistem, maka sistem siap untuk diimplementasikan. Pada tahap kontruksi dilakukan pembangunan dan menguji sebuah sistem fungsional yang memenuhi persyaratan bisnis dan desain untuk diimplementasikan antarmuka antara sistem baru dan lama (Whitten, Bentley and Dittman, 2004).

8. *Installation and Delivery*

   Pada tahap ini akan dioperasikan sistem yang telah dibangun. Tahapan ini akan dimulai dengan *development* software hingga memberikan pelatihan kepada user mengenai penggunaan sistem yang telah dibangun. Pada proses instalasi harus memperhatikan instalasi dari perangkat lunak yang telah dibuat serta instalasi *database* (Whitten, Bentley and Dittman, 2004).

Salah satu metodologi pengembangan sistem yang umum dipakai adalah metodologi FAST (*Framework for the Application of Systems Technique*). FAST dikembangkan sebagai gabungan dari praktek-praktek terbaik yang telah ditemui dalam banyak referensi komersial dan metodologi. Metodologi FAST hipotesis tidak menggunakan pendekatan tunggal pada analisis sistem, akan tetapi mengintegrasikan semua pendekatan populer yang diperkenalkan dalam satu kumpulan (*agile methode*). Berikut ini adalah gambaran pembagian perancangan sistem informasi menggunakan metode FAST (Whitten & Bentley, 2007).

2.10. *Entity/ Relationship Diagram* (Diagram E/R)

Pemodelan awal basis data yang align digunakan adalah mengunakan *entity relationship diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan

dalam bidang matematika (Rosa A.S dan M.Shalahuddin, 2016). Ada sejumlah konvensi mengenai Notasi ERD. Notasi klasik sering digunakan untuk model konseptual. Notasi-notasi simbolik yang digunakan dalam Entity Relationship Diagram adalah sebagai berikut :

1. Entitas, Adalah segala sesuatu yang dapat digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain (Hoffer, 2009). Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya. Contohnya entitas anggota. Sedangkan entitas lemah merupakan entitas yang kemunculannya tergantung pada keberadaaan entitas lain dalam suatu relasi.
2. Atribut, Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau elips. Atribut yang menjadi kunci entitas atau key diberi garis bawah.
3. Relasi atau Hubungan, Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.
4. Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut dinyatakan dalam bentuk garis

Kardinalitas Menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain (Hoffer, 2009). Macam-macam kardinalitas adalah:

a. Satu ke satu (*one to one*), Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.
b. Satu ke banyak (*one to many*), Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.
c. Banyak ke banyak (*many to many*), Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.