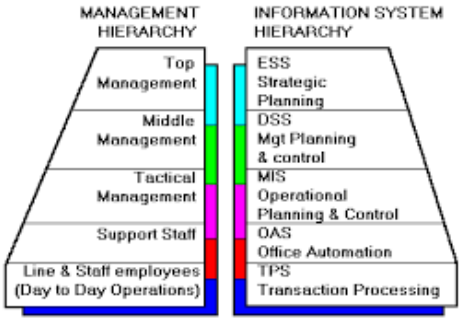
Bab 2
Tinjauan Pustaka

2.1. Sistem Informasi

Sebuah sistem adalah sekelompok komponen yang saling terkait yang berfungsi bersama-sama untuk mencapai hasil yang diinginkan. Suatu sistem informasi (IS) adalah pengaturan orang, data, proses, dan teknologi informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan, dan menyediakan sebagai *output* informasi yang dibutuhkan untuk mendukung sebuah organisasi. Sedangkan Teknologi informasi adalah istilah kontemporer yang menggambarkan kombinasi teknologi komputer (*hardware dan software*) dengan teknologi telekomunikasi (data, gambar, dan jaringan suara) (Whitten, Bentley and Dittman, 2004).



Gambar 2.1. Hirarki Manajemen dan Sistem Informasi
(Whitten, Bentley and Dittman, 2004).

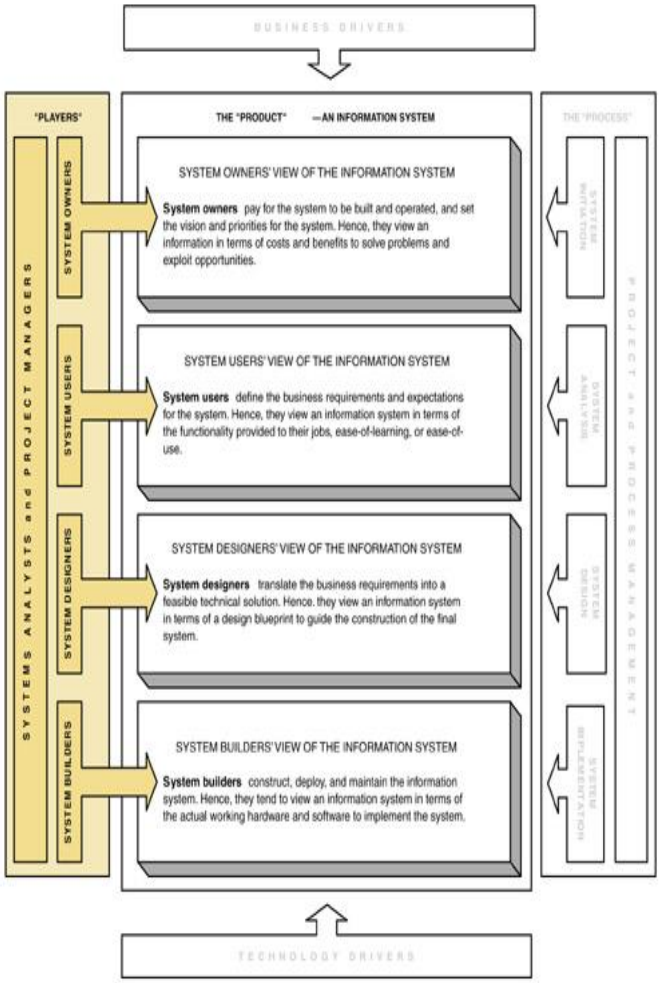
Berikut ini adalah penjabaran dari hirarki sistem informasi:

1. Sebuah sistem pemrosesan transaksi (TPS) adalah sistem informasi yang menangkap dan memproses data tentang transaksi bisnis.
2. Sebuah sistem informasi manajemen (MIS) adalah sistem informasi yang menyediakan untuk pelaporan manajemen berorientasi berdasarkan pemrosesan transaksi dan operasi organisasi.

3. Sebuah sistem pendukung keputusan (DSS) adalah sistem informasi yang baik membantu untuk mengidentifikasi peluang pengambilan keputusan atau menyediakan informasi untuk membantu membuat keputusan.
4. Sistem pakar adalah suatu sistem informasi yang menangkap keahlian pekerja dan kemudian mensimulasikan keahlian yang untuk kepentingan non-ahli.
5. Sebuah sistem komunikasi dan kolaborasi adalah sistem informasi yang memungkinkan komunikasi lebih efektif antara pekerja, mitra, pelanggan, dan pemasok untuk meningkatkan kemampuan mereka untuk berkolaborasi.
6. Sebuah sistem otomatisasi kantor adalah sistem informasi yang mendukung berbagai kegiatan kantor bisnis yang menyediakan alur kerja ditingkatkan antara pekerja.

2.1.1. Stakeholder Sistem Informasi

Sebuah pemangku kepentingan (*Stakeholder*) adalah setiap orang yang memiliki kepentingan dalam suatu sistem informasi yang ada atau yang diusulkan. Stakeholder dapat tenaga teknis atau nonteknis. Mereka juga mungkin termasuk pekerja baik internal maupun eksternal. Pekerja informasi adalah mereka pekerja yang pekerjaannya melibatkan penciptaan, pengumpulan, pengolahan, distribusi, dan penggunaan informasi. Pekerja pengetahuan adalah bagian dari pekerja informasi yang tanggung jawab didasarkan pada tubuh khusus pengetahuan (Whitten, Bentley and Dittman, 2004).



Gambar 2.2. *Stakeholders' Perspectives on an Information System*
(Whitten, Bentley and Dittman, 2004).

Berikut ini adalah *Stakeholders* yang terlibat dalam perangcangan sistem informasi:

1. Pemilik sistem (*System Owner*)

   Sistem informasi ini sponsor dan advokat eksekutif, biasanya bertanggung jawab untuk pendanaan proyek pengembangan, operasi, dan memelihara sistem informasi.

2. Pengguna sistem (*System User*)

   "Pelanggan" yang akan menggunakan atau dipengaruhi oleh suatu sistem informasi secara teratur - menangkap, memvalidasi, memasukkan, merespon, menyimpan, dan pertukaran data dan informasi. Berikut ini adalah pengguna sistem:

a. Pekerja administrasi dan layanan
b. Staf teknis dan profesional
c. Supervisor, manajer menengah, dan manajer eksekutif

3. Sistem desainer (*System Designer*)

   Spesialis teknis yang menerjemahkan kebutuhan bisnis pengguna sistem dan kendala menjadi solusi teknis. Dia atau dia desain database komputer, *input*, *output*, layar, jaringan, dan perangkat lunak yang akan memenuhi kebutuhan pengguna sistem. Berikut ini adalah orang-orang yang terlibat dalam sistem desainer (Whitten, Bentley and Dittman, 2004).:

   a. Database administrator-spesialis dalam terminologi database yang merancang dan mengkoordinasikan perubahan pada database perusahaan.
   b. Arsitek jaringan - spesialis di bidang jaringan dan telekomunikasi yang mendesain, Menginstal, mengkonfigurasi, mengoptimalkan, dan mendukung jaringan area lokal dan luas, termasuk koneksi ke lntemet dan jaringan eksternal lainnya.
   c. Arsitek Web - spedalis yang merancang situs Web yang kompleks untuk organisasi, Menunjukkan situs Web publik untuk lntemet, situs Web Internal untuk organisasi (lntranet), dan situs web bisnis-ke-bisnis swasta (ekstranet).

4. Pembangun sistem (*System Builder*)

   Spesialis teknis yang membangun sistem informasi dan komponen berdasarkan spesifikasi desain yang dihasilkan oleh desainer sistem (Whitten, Bentley and Dittman, 2004).

   a. Ahli program aplikasi - spesialis yang mengubah persyaratan bisnis dan laporan masalah dan prosedur Ke dalam bahasa komputer. Mereka mengembangkan dan mengistirahatkan program komputer untuk menangkap dan menyimpan data dan mencari dan mengambil data untuk aplikasi komputer.
   b. Pemrogram sistem - spesialis yang mengembangkan, menguji, dan menerapkan perangkat lunak, utilitas, dan layanan tingkat operasi. Semakin

banyak, mereka juga mengembangkan perangkat lunak yang dapat digunakan kembali "komponen" untuk digunakan oleh aplikasi progammers (di atas).

c. Database programmer - spesialis dalam bahasa database dan teknologi yang membangun, memodifikasi, dan menguji struktur database dan program yang menggunakan dan merawatnya.
d. Administrator jaringan-spesialis yang merancang, menginstal, memecahkan masalah, dan mengoptimalkan jaringan komputer.
e. Administrator keamanan-spesialis yang merancang, menerapkan pemecahan masalah dan mengelola kontrol keamanan dan privasi dalam jaringan .

2.2. Perangkat Lunak

Perangkat lunak (*software*) merupakan sebuah program komputer yang terasosiasi dengan dokumentasi perangkat lunak. Berikut ini adalah dokumentasi dari perangkat lunak (Rosa & Shalahuddin, 2015):

1. Dokumentasi kebutuhan
2. Dokumentasi model desain
3. Dokumentasi cara penggunaan (*manual book*)

Sistem perangkat lunak adalah sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain dan berfungsi untuk memenuhi kebutuhan pelanggan (orang maupun organinasi) yang menggunakan perangkat lunak dengan cara memesan atau membeli perangkat lunak. Berikut ini adalah karakter dari perangkat lunak itu sendiri:

1. Perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi melalui proses manufaktur atau pabrikan
2. Perangkat lunak tidak memiliki batas waktu atau usang (*wear out*) karena kecacatan dalam perangkat lunak dapat diperbaiki
3. Perangkat lunak akan terus diperbaiki seiring bertambahnya kebutuhan

Adapun aplikasi perangkat lunak adalah sebagai berikut (Rosa & Shalahuddin, 2015):

1. Perangkat lunak sistem (*system software*)

   System software merupakan kumpulan dari program yang ditulis untuk memenuhi kebutuhan program lainnya.
2. Perangkat lunak waktu nyata (*real-time software*)

   Real-time software merupakan perangkat lunak yang memonitor, meganalisis, mengontrol sesuatu secara nyata (*real-time*).
3. Perangkat lunak bisnis (*business software*)

   Business software adalah perangkat lunak yang berfungsi sebagai pengelola informasi bisnis.
4. Perangkat lunak untuk keperluan rekayasa dan keilmuan (*engineering and scientif software*)

   Engineering and scientif software adalah perangkat lunak yang mengimlementasikan algoritma yang terkait dengan keilmuan ataupun perangkat lunak yang membantu keilmuan, misalkan perangkat lunak di bidang astronomi dan sebagainya.
5. Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak lainnya (*embedded software*).

   Embedded software merupakan perangkat lunak tambahan misalnya perangkat lunak mencatak dokumen ditambahkan agar perangkat lunak yang memerlukan dapat mencetak laporan.
6. Perangkat lunak komputer personal (*personal computer software*)

   Personal computer software adalah perangkat lunak untuk Komputer seperti perangkat lunak pemroses teks, grafis dan lainnya.
7. Perangkat lunak berbasis web (*web based software*)

   Web based software merupakan perangkat lunak yang dapat diakses dengan menggunakan browser.

2.2.1. Rekayasa perangkat lunak

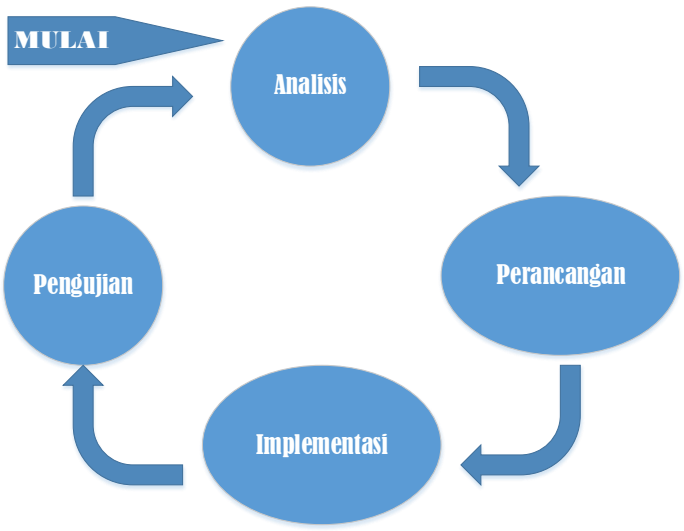
Software engineering adalah pembangunan perangkat lunak dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomis dan mampu bekerja secara efisien menggunakan mesin .Rekayasa perangkat lunak lebih fokus pada pembangunan dan mengirimkan perangkat 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.

2.2.2. 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.

2.3. 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

2.4. 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.

2.5. 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.*

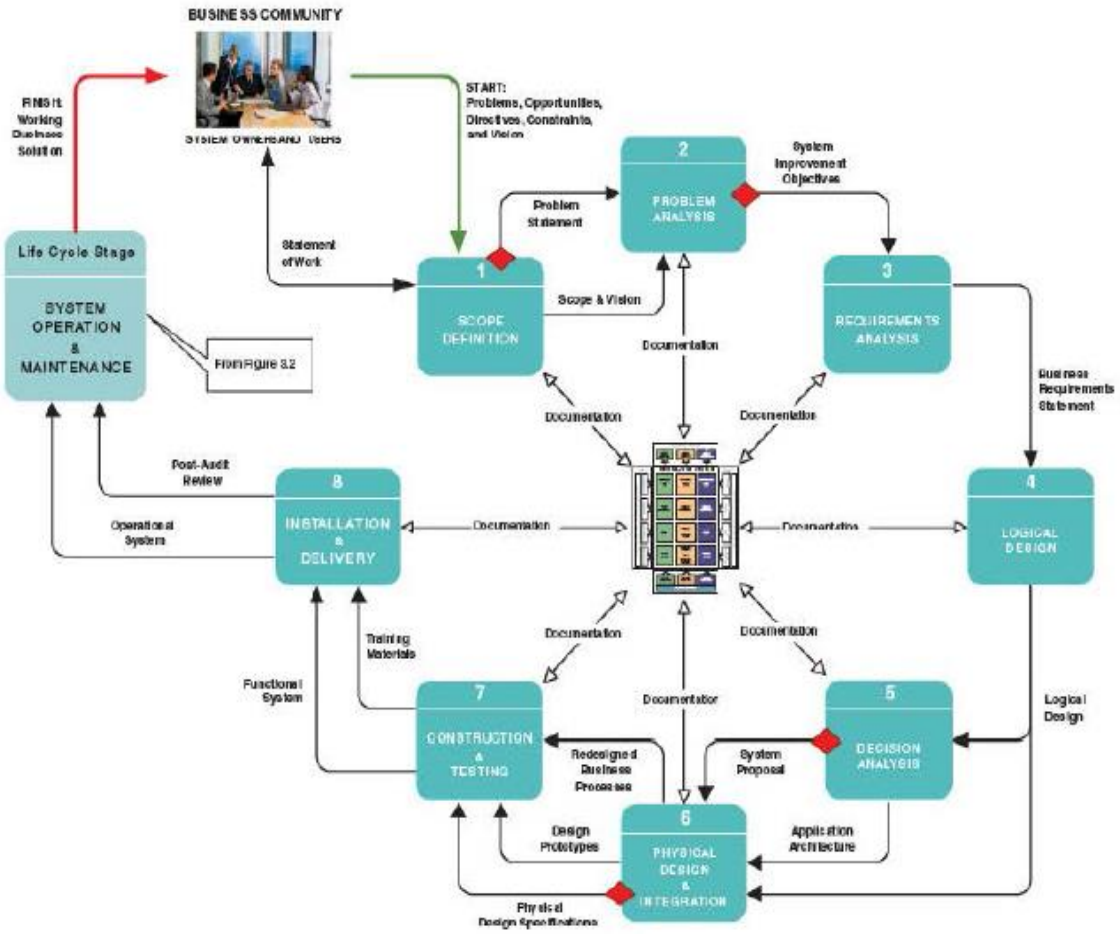
Analisis terstruktur adalah salah satu pendekatan formal pertama untuk analisis sistem informasi dengan berpusat pada proses yang digunakan untuk menganalisis sistem yang ada, mendefinisikan persyaratan-persyaratan bisnis untuk sebuah sistem baru. Teknik informasi adalah suatu pendekatan yang berpusat 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).

2.5.1. 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).

FAST Phases	Classic Phases			
	Project Initiation	System Analysis	System Design	System Implementation
Scope definition	X			
Problem analysis	X	X		
Requirements analysis		X		
Logical design		X		
Decision analysis	(a system analysis transition phase)			
Physical design and integration			X	
Construction and testing			X	X
Installation and delivery				X

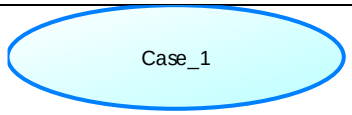
Gambar 2.5. Pembagian Perancangan Sistem Informasi Sistem FAST (Whitten & Bentley, 2007).

2.6. Pemodelan *Use-Case*

Use-case modeling merupakan suatu proses pemodelan fungsi-fungsi sistem dalam konteks peristiwa-peristiwa bisnis, siapa yang mengawalinya dan bagaimana sistem itu merespon hal tersebut. Ada dua alat utama yang digunakan saat menyajikan pemodelan *use-case* yaitu *use-case diagram* dan *use-case narrative*. *Use-case diagram* merupakan diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna, sedangkan *use-case narrative* merupakan deskripsi tekstual kegiatan bisnis dan bagaimana pengguna akan berinteraksi dengan sistem untuk suatu tugas. *Use-case* sendiri merupakan urutan langkah-langkah yang secara tindakan saling terkait, baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal (Whitten, Bentley and Dittman, 2004).

Adapun syarat penamaan pada *use-case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami, serta ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use-case* (Rosa & Shalahuddin, 2015):

Tabel 2.2. Pendefinisian *Use-case*

Simbol	Deskripsi
 Aktor	Aktor merupakan orang ataupun sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat
 *Use-Case*	*Use-case* merupakan fungsionalitas yang disediakn sistem sebagai unit-unit yang salng bertukar pesan antar unit atau aktor.

Pada diagram *use-case* terdapat hubungan yang digambarkan sebagai sebuah garis antara aktor dan *use-case*. *Association* (Gabungan) merupakan hubungan antara seorang aktor dan satu *use-case* terbentuk. *Extend* merupakan hubungan *use-case* yang terdiri dari langkah yang diekstraksi dari *use-case* yang lebih kompleks untuk menyederhanakan masalah orisinal. *Abstract use-case* merupakan *use-case* yang menggabungkan anatara dua atau lebih *use-case* lain dengan menggabungkan langkah-langkah yang biasa ditemukan pada *use-case* tersebut. *Depends On* merupakan hubungan *use-case* yang memodelkan ketergantungan dari *use-case* sistem. *Include* merupakan hubungan *use-case* tambahan ke sebuah *use-case* di mana *use-case* yang ditambahkan memerlukan *use-case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use-case* (Whitten, Bentley and Dittman, 2004).

2.7. Basis Data

Sistem basis data merupakan sistem yang terkomputerisasi serta memiliki tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Intinya basis data adalah media untuk menyimpan data agar dapat di akases dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya baik

berupa *file text* maupun *database management system* (DBMS). Kebutuhan basis data dalam sistem informasi melitputi (Rosa & Shalahuddin, 2015):

1. Kebutuhan memasukan, menyimpan dan mengambil data.
2. Kebutuhan membuat laporan berdasarkan data yang telah disimpan.

2.7.1. *Database Management System* (DBMS)

DBMS adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi syarat berikut (Rosa & Shalahuddin, 2015):

1. Mampu menyediakan fasilitas untuk mengelola akses data.
2. Mempu menangani integritas data
3. Mampu menangani akses data
4. Mampu menangani *backup* data

Hampir sebagaian besar perusahaan memanfaatkan DBMS dalam mengelola data, maka daripada itu DBMS dianggap penting bagi suatu organisasi/perusahaan. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli yang disebut sebagai DBA (*Database Administrator*).

2.7.2. Alur Hidup Basis Data

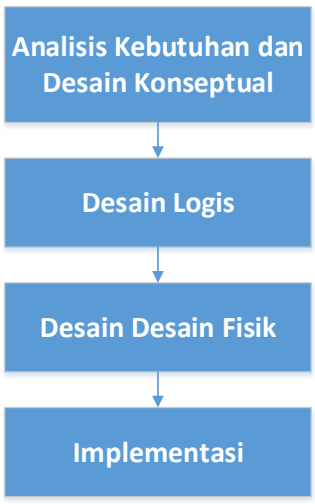
Perencanaan basis data juga memiliki alur hidup atau sering disebut dengan *Database Life Cycle* (DBLC). Terdapat 4 Fase DBLC antara lain (Rosa & Shalahuddin, 2015):

1. Analisis kebutuhan/requirement analisis
2. Desain logis basis data, pada tahap ini dibuat *conceptual data model* (CDM). *Conceptual data model* merupakan sebuah konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi basis data.
3. Desain fisik basis data, pada tahap ini dibuat *physical data model* (PDM). *Physical data model* merupakan model yang menggunakan sejumlah tabel untuk

menggambarkan data serta hubungan antar data. PDM merupakan konsep yang menerangkan detail dari bagaimana data di simpan di dalam basis data.

4. Implementasi

Berikut ini adalah gambaran DBLC dapat dilihat pada gambar 2.6.



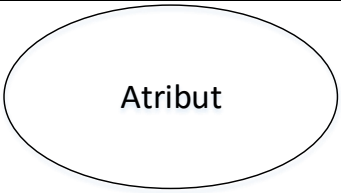
Gambar 2.6. Alur Hidup Basis Data
(Rosa & Shalahuddin, 2015)

2.8. Konsep Sistem

2.8.1. Konsep Sistem Pemodelan Data

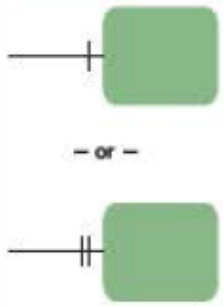
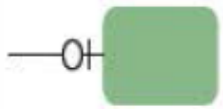
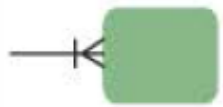
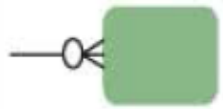
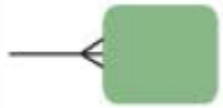
Pemodelan awal basis data yang paling banyak digunakan adalah *Entity Relationship Diagram* (ERD). ERD merupakan model data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang dideskripsikan oleh data tersebut. Berikut ini adalah simbol simbol yang digunakan pada ERD dengan notasi Chen dapat dilihat pada tabel 2.3. (Whitten, Bentley and Dittman, 2004)

Tabel 2.3. Simbol ERD Notasi Chen

Simbol	Deskripsi
	Entias adalah sesuatu yang diperlukan bisnis untuk menyimpan data.
	Atribut atau data merupakan karateristik deskriptif dari suatu entitas.
	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

2.8.2. Hubungan (*Relationship*)

Relationship merupakan hubungan bisnis alami yang ada antara satu entitas atau lebih entitas, hubungan tersebut dapat menyatakan kejadian yang menghubungkan entitas atau hanya persamaan logika yang ada di antara entitas. *Cardinality* merupakan jumlah minimum dan maksimum kemunculan satu entitas yang mungkin di hubungkan dengan kemunculan tunggal dari entitas lain. Berikut ini adalah gambaran dari kardinalitas suatu relasi, dapat dilihat pada gambar 2.7 (Whitten, Bentley and Dittman, 2004)

CARDINALITY INTERPRETATION	MINIMUM INSTANCES	MAXIMUM INSTANCES	GRAPHIC NOTATION
Exactly one (one and only one)	1	1	
Zero or one	0	1	
One or more	1	many (>1)	
Zero, one, or more	0	many (>1)	
More than one	>1	>1	

Gambar 2.7. Notasi Kardinalitas

(Whitten, Bentley and Dittman, 2004)

2.8.3. Konsep Sistem Pemodelan Proses

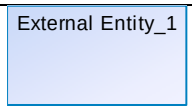
Proses adalah kerja yang dilakukan oleh sistem sebagai respon terhadap aliran data masuk atau kondisi. Dekomposisi adalah kegiatan menguraikan sistem menjadi subsistem, proses dan subproses komponennya. Dekomposisi diagram juga pada dasarnya merupakan alat perencanaan yang digunakan untuk model proses yang lebih detail, yang disebut dengan diagram aliran data.

Proses logika adalah pekerjaan atau tindakan yang harus dilakukan dalam mengimplementasikan sistem, terdapat tiga proses logika yaitu (Whitten, Bentley and Dittman, 2004):

1. Fungsi merupakan kumpulan kegiatan yang saling terkait dan terus menerus pada suatu bisnis. Fungsi tidak memiliki awal dan akhir, melainkan bekerja terus-menerus pada saat dibutuhkan.
2. Kejadian merupakan unit kerja logika yang harus diselesaikan secara keseluruhan. Suatu kejadian dipicu oleh *input* diskrit dan diselesaikan pada proses yang merespon kejadian.
3. Proses elementer merupakan kegiatan atau tugas diskrit dan detail yang dibutuhkan untuk merespon suatu kejadian.

Data flow diagram (DFD), dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam Structured Systems Analisys and Design Methodology (SSADM). Pada tahun 1980 Edward Yourdin dan Tom DeMarco meperkenalkan metode lain dengan mengubah persegi dengan sudut dengan lingkaran untuk menotasikan. DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Berikut ini notasi yang digunakan pada DFD yang dikembangkan oleh Chris Gane & Trish Sarson dapat dilihat pada tabel 2.4.

Tabel 2.4. Notasi DFD Chris Gane & Trish Sarson

Notasi	Deskripsi
	*External Entitiy* mendefinikan orang unit organisasi, sistem atau organisasi luar yang berinteraksi dengan sistem
	*Data store* mendefinisikan inventory data

2.9. SDLC (*Software Development Life Cycle*)

Software Development Life Cycle atau sering disebut juga dengan *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut (Rosa & Shalahuddin, 2015):

1. Inisiasi, pada tahap ini biasanya ditandai dengan adanya pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem yaitu mendefinisikan lingkup konsep yang termasuk kedalam dokumen lingkungan sistem, analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.
3. Perencanaan adalah dengan mengembanganken rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya yang dibutuhkan dalam pencarian solusi .
4. Analisis kebutuhan yaitu menganalisis setiap kebutuhan dari pemakai sistem perangkat lunak dan mengembangkan kebutuhan *user*.
5. Desain yaitu mentransformasikan kebutuhan secara detail dari hasil analisis kebutuhan secara lengkap.
6. Pengembangan yaitu kegiatan mengonverikan desain ke dalam sistem infomasi yang lengkap, termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan.
7. Integrasi dan pengujian yaitu mendemontrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan pada kebutuhan fungsional.
8. Kegiatan mengimplementasikan perangkat lunak pada lingkungan dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.
9. Operasi dan pemeliharaan yaitu endeskripsikan pekerjaan untuk mengoprasikan dan memelihara sistem informasi pada lingkungan, termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi yaitu mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

2.9.1. Model SDLC

SDLC memiliki beberapa model dalam penerapan tahapan-tahapan dari prosesnya, berikut ini adalah beberapa model dari SDLC (Rosa & Shalahuddin, 2015):

1. Model *Waterfall*

   Model ini sering disebut juga model sekuensial linier atau alur hidup klasik, model air terjun memiliki alur sebagai berikut:

   a. Analisis kebutuhan perangkat lunak

      Proses pengumpulan kebutuhan-kebutuhan dilakukan secara intensif hal ini dilakukan untuk mengspesifikasikan kebutuhan dari perangkat lunak agar mampu dipahami seperti yang dibutuhkan oleh *user*. Pada tahap ini spesifikasi kebutuhan perangkat lunak perlu untuk di dokumentasikan.

   b. Desain

      Desain perangkat lunak merupakan suatu proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean.

   c. Pembuatan kode program.

      Hasil dari tahap ini adalah program komputer yang telah dibuat berdasarkan tahap desain.

   d. Pengujian

      Proses pengujian fokus pada perangkat lunak secara dari logika dan fungsional dan memastikan bahwa semua bagian sudah di uji.

   e. Pemeliharaan

      Perubahan yang bisa muncul karena adanya kesalahan dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

2. Model Prototipe

   Model prototipe dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Model ini dimulai dengan mengumpulakan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat, setelah itu dibuat program prototipe agar pelanggan dapat terbayang program yang diinginkan. Seiring dengan mengembangkan prototipe maka sistem perangkat lunak yang sebenarnya dikembangkan juga, sehingga sesuai dengan kebutuhan pelanggan.

   Model prototipe cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan menyampaikan kebutuhan tanpa melihat gambaran yang jelas, namun sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembangan dengan pelanggan secara tertulis. Maka dokumen tersebut akan menjadi patokan agar spesifikasi kebutuhan sistem masih dalam ruang lingkup proyek. Model prototipe kurang cocok untuk aplikasi dengan skala yang besar karena akan memakan waktu dan tenaga.

2.10. Desain Sistem

Desain sistem informasi didefinisikan sebagai proses yang fokus pada spesifikasi solusi detail berbasis komputer dan dapat juga disebut sebagai desain fisik. Ada dua tujuan utama fase desain yaitu analisis mendesain sebuah sistem baik yang memenuhi persyaratan dan analisis akan menunjukan spesifikasi yang lengkap dan jelas pada programer dan teknisi komputer. Strategi pada desain sistem FAST tidak memaksakan sebuah pendekatan tunggal pada berbagai desain sistem (Whitten, Bentley and Dittman, 2004).

Tugas pertama pada desain sistem adalah dengan untuk menentukan sebuah arsitektur aplikasi. Tugas ini dapat diselesaikan dengan cara menganalisis model data dan proses yang pada awalnya dibuat pada analisis persyaratan. Contohnya

adalah dengan menggunakan *physcal data flow diagram*. Diagram aliran data fisik digunakan untuk memodelkan keputusan-keputusan teknis dan desain untuk diimplementasikan sebagai bagian dari sistem informasi.

Tugas selanjutnya adalah dengan mendesain *database systems* dengan cara mengembangkan spesifikasi desain database yang sesuai. Desain dapat berkembang terus menerus sehingga harus diperhatikan tata letak dari record. *Database* adalah sumber yang digunakan secara bersama-sama sehingga harus didesain dengan baik supaya dapat beradaptasi dengan persyaratan dan pengembangan pada masa yang akan datang.

Tugas ketiga dari desain sistem adalah mendesain antarmuka sistem, setelah *database* di desain maka langkah selanjutnya adalah dengan membuat prototipe atau desain awal untuk mengembangkan *input*, *output* dan spesifikasi dialog. Pada antar muka *input* maka dapat dilakukan dengan mendesain sebuah form dimana data yang akan di *input* terekam. Pada desain antarmuka juga harus mempertimbangkan beberapa faktor seperti familiaritas terminal, error dan kesalahan yang mungkin terjadi.

2.11. Kontruksi dan Implementasi Sistem

Kontruksi sistem merupakan sebuah pengembangan, instalasi dan pengujian pada komponen-komponen sistem, sedangkan implementasi sistem adalah pengiriman sistem untuk digunakan sehari-hari. Pemicu fase kontruksi adalah persetujuan spesifikasi desain fisik yang merupakan hasil dari fase desain. Berikut ini adalah kegiatan-kegiatan yang dilakukan pada fase kontruksi (Whitten, Bentley and Dittman, 2004):

1. Membangun dan mengetes jaringan.
2. Membangun dan mengetes *database*.
3. Menginstal dan mengetes perangkat lunak.
4. Menulis dan mengetes program baru.

Pada tahap implementasi tugas yang dilakukan adalah sebagai berikut:

1. Menguji sistem.
2. Menyiapkan rencana konversi.
3. Menginstal *database.*
4. Melatih para pengguna.
5. Mempersiapkan diri berarlih ke sistem yang baru.