BAB II
TINJAUAN PUSTAKA

2.1 Sistem Informasi

2.1.1 Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengna laporan-laporan yang dibutuhkan[3].

2.1.2 Konsep Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu [3]:

1. Blok masukkan (*input block*)

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

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prsedur, logika dan metode matematik yag akan memanipulasi data input dan data yang tersimmpan di basis data dengna cara yang sudah tertentu untuk menghasilkna keluaran yang sudah diinginkan.

3. Blok keluaran (*output block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technologi block*)

   Teknologi digunakan utnk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan. Teknologi terdiri dari unsur utama:

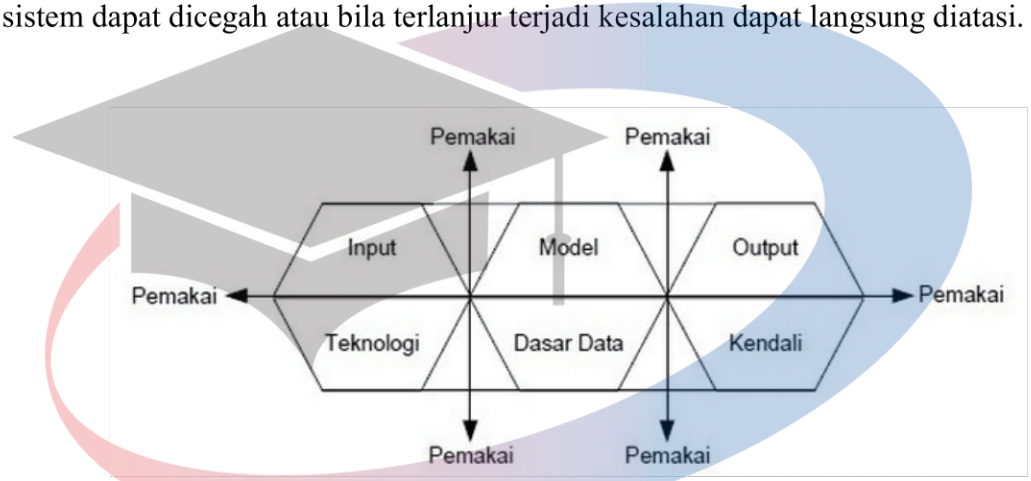
   a. Teknisi (*human ware* atau *brain ware*)
   b. Perangkat lunak (*software*)
   c. Perangkat keras (*hardware*)

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

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

6. Blok kendali (*control block*)

Banyak factor yang dapat merusak sistem informasi, misalnya bencana alama, api, temperature tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.



Gambar 2.1 Blok Sistem Informasi yang Berinteraksi

2.2 Rekayasa Perangkat Lunak

2.2.1 Pengertian Rekayasa Perangkat Lunak

Rekayasa atau teknik adalah penerapan ilmu dan teknologi untuk menyelesaikan permasalahan manusia. Rekayasa perangkat lunak (RPL atau SE [*Software Engineering*]) adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak, dan sebagainya. Biasanya proses melibatkan penemuan pada keinginan klien, menyusunnya di dalam daftar kebutuhan, perangan, pengodean, pengujian, dan pengintegrasian bagian yang terpisah, menguji keseluruhan, penyebaran dan pemeliharaan perangkat lunak. Pemrograman hanya menjadi bagina kecil dari rekayasa perangkat lunak[4].

The Software Engineering Body of Knowledge (SWEBOK) membagi rekayasa perangkat lunak ke dalam 10 area pengetahuan, yaitu:

1. Kebutuhan perangkat lunak,
2. Perancangan perangkat lunak,
3. Konstruksi perangkat lunak,

4. Pengujian perangkat lunak,
5. Pemeliharaan perangkat lunak,
6. Manajemen konfigurasi perangkat lunak,
7. Manajemen perangkat lunak,
8. Proses perangkat lunak,
9. Metode dan tool perangkat lunak, dan
10. Kualitas perangkat lunak.

2.2.2 Jenis Model Proses Rekayasa Perangkat Lunak

Dasar untuk rekayasa perangkat lunak adalah lapisan proses. Proses rekayasa perangkat lunak adalah proses yang terus berulang, karena karakteristik perangkat lunak yang membutuhkan pemeliharaan dan pengembangan berkelanjutan agar perangkat lunak tidak kadarluarsa.

Pekerajaan yang berhubungan dengan rekayasa perangkat lunak dapat dikategorikan ke dalam tiga fase generik, yaitu[4]:

1. Tahap definisi

   Berfokus pada *what*. Pada fase ini mengidentifikasi informasi apa yang akan diproses, apa fungsi dan kinerja yang diinginkan, perilaku sistem apa yang dapat diharapkan, apa antarmuka yang akan didirikan, apa desain kendala yang ada, dana pa kriteria validasi yang diperlukan untuk menentukan sistem yang sukses. Persyaratan utama dari sistem dan perangkat lunak diidentifikasi. Tiga tugas utama akan terjadi dalam beberapa bentuk: sistem atau teknik informasi, perencanaan proyek perangkat lunak, dan analisis kebutuhan.

2. Tahap pengembangan

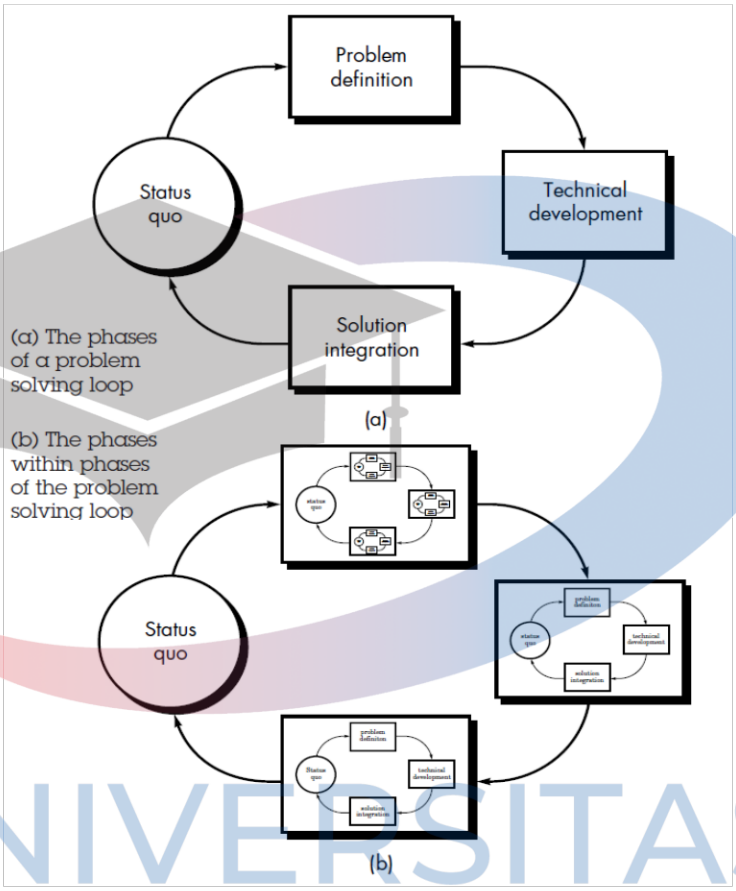
   Berfokus pada *how*. Selama pengembangan perangkat lunak didefinisikan bagaimana data harus terstruktur, bagaimana fungsi diimplementasikan dalma arsitektur, perangkat lunak, bagaimana detail procedural untuk dilaksanakan, bagaimana interface yang akan ditandai, bagaimana desain akan diterjemahkan ke dalam Bahasa pemrograman (atau Bahasa nonprocedural), dan bagaimana pengujian akan dilakukan. Metode yang diterapkan dalma tahap pengembangan akan bervariasi, tetapi tiga tugas teknis tertentu harus selalu terjadi: desain perangkat lunak generasi kode, dan pengujian perangkat lunak.

3. Fase dukungan

   Berfokus pada perubahan yang terkait dengan koreksi kesalahan.

2.2.3 Model Proses dalam Rekayasa Perangkat Lunak

Sebuah model proses rekayasa perangkat lunak dipilih berdasarkan pada sifat proyek dan aplikasi, metode dan alat-alat yang akan digunakan, dan control dan kiriman yang diperlukan[4].



Gambar 2.2 Lingkaran Fase Pemecahan Masalah

2.3 Pengujian Perangkat Lunak

2.3.1 Pengertian Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses untuk menemukan kesalahan sebelum di kirim kepada pengguna. Pengujian perangkat lunak merupakan kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean Pentingnya pengujian perangkat lunak adalah untuk dapat menjalankan program dengan maksud mencari kesalahan, dan kasus uji yang baik yaitu kasus yang memiliki peluang untuk mendapatkan kesalahan yang belum diketahui. Pengujian dikatakan berhasil apabila dapat memunculkan kesalahan yang belum diketahui, dan pengujian yang baik bukan untuk

memastikan tidak adanya kesalahan, tetapi untuk mencari sebanyak mungkin kesalahan yang ada pada program[5].

Sebuah perangkat lunak perlu dijaga keualitasnya bahwa kualitas bergantung kepada kepuasan pelanggan (*customer*). Kualitas perangkat lunak perlu dijaga untuk keperluan sebagai berikut [6]:

a. Agar dapat "*survive*" bertahan hidup di dunia bisnis perangkat lunak
a. Dapat bersaing dengan perangkat lunak yang lain
b. Penting untk pemasaran global (*global marketing*)
c. Mengefektifkan biaya agar tidak banyak membuang perangakt lunak karena kegagalan pemasaran atau kegagalan produksi
d. Mempertahankan pelanggan (*customer*) dan meningkatkan keuntungan

Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desainkasus uji yang spesifitik dan metode pengujian. Secara umum pola pengujian pada perangkat lunak sebagai berikut[6]:

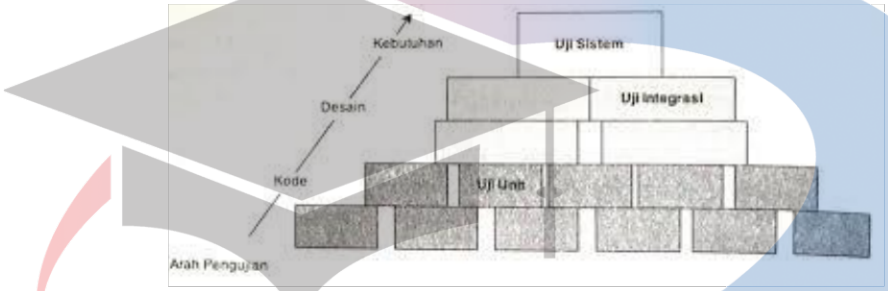
1. Pengujian dimulai dari level komponen hingga integrase antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengna berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujiian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengmbang perangkat lunak (*independet test group* (ITG)).
4. Pengujian dan penirkutuan (*debugging*) merupakan aktifitas yang berbeda, tapi pernikutuan (*debugging*) harus diakomodasi pada berbagai strategi pengujian. Pengujian lebih focus untuk mencari adanya kesalahan (*error*) baik dari sudut pandang orang secara umum atau dari sudut pandang pengmbang tanpa harus menemukan lokasi kesalahan pada kode program. Penirkutuan (*debugging*) adalah proses mencari lokasi kesalahan (*error*) pada kode program sehingga segera diperbaiki oleh pembuat program (*programmer*).

Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengna verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengna benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangakt lunak yang dibangun

dapat ditelursuri sesuai dengan kebutuhan pelangan (*customer*). Dapat juga dikatakan sebagai berikut:

a. **Verifikasi**: "Apakah produk dibangun dengan benar?" (lebih ke arah apakah proses pengembangan produk sudah benar dan telah berhasil mengimplementasikan fungsi yang benar)
b. **Validasi**: "Apakah sudah membangun produk yang benar?" (lebih ke arah hasil produk apakah sudah sesuai dengan yang diinginkan)

Pengujian untuk verifikasi dilakukan dimulai dari lingkup yang kecil naik ke lingkup yang besar seperti pada gambar berikut[6]:



Gambar 2.3 Hirarki Pengujian Sistem

Gambar di atas menunjukkan tahap pengujian pada level program di tangan pengembang perangkat lunak. Tahapan pengujia yang secara keseluruhan adalah sebagai berikut:

Gambar 2.4 Pengujian Perangkat Lunak

Pengujian diawali dari pengujian unit. Unit disini bisa berupa kumpulan fungsi atau prosedur yang memiliki keterkaitan pada pemrograman terstruktur (misalkan unit untuk menuliskan atau membaca data di basis data) atau kelas pad apemrograman berorientasi objek. Unit juga dapat berupa modul atau dikenal juga sebagai *package*. Setelah unit-unit selesai diuji maka dilakukan pengujian integrasi.

Pengujian integrase sebaiknya dilakukan secra bertahap, tidak dilakukan secara satu tahap langsung di akhir untuk menghidari kesulitan penelusuran jika terjadi kesalahan (*error*). Pengujian integrase lebih pada pengujian penggabungan dari dua atau lebih unit pada perangkat lunak. Setelah pengujian integrase maka dilakukan pengujian sistem dimana unit-nit proses yang sudah diintegrasi diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksudkan untk menguji sistem perangkat lunak secara keseluruhan dan diuji secara satu sistem (tidak terpisah-pisah lagi).

2.3.2 Manajemen Pengujian Perangkat Lunak

Dalam mengangkat konsep-konsep pengujian perangkat lunak kepada suatu pengimplementasian, manajemen yang mengatur keseluruhan proses pengujian akan diperluakan. Proses manajemen pengujian perangkat lunak menangani berbagai aktivitas dan langkah-langkah pengujian perangkat lunak, di antranya adalah inisialisasi dan penentuan lingkup kerja pengujian, termasuk di dalamnya batasan-batasan pengujian serta lingkungan pengujian, perencanaan pengujian perangkat lunak, eksekusi dan pengendalian perangkat lunak selama pengujian, evaluasi pengujian serta peninjauan ulang hasil pengujian, dan penutup, yaitu pelaporan hasil pengujia yang diikuti dengna pembenahan bila diperlukan. Beberapa fakta di lapangan tentang pengujian perangk lunak, antara lain[4]:

1. Instansi-instansi penguji tidak memandang perlunya gelar pendidikan tertentu pada penguji penguji perangkat lunak, melainkan sertifikasi-sertifikasi pengujian serta pengalaman yang menunjukkan kapabilitas seorang penguji perangkat lunak. Dalam hal ini, perusahaan tidak perlu harus membayar mahal untuk menyesuaikan gelar-gelar penguji, namun tetap mendapatkan hasil yang memuaskan dengan kredibitas penguji.
2. Proses pengujian memiliki kecenderungan untuk memand perangkat lunak dari sisi kebutuhan pengguna sehingga takaran yang diukur merupakan kesesuaian dan kemampuan perangkat lunak untuk memenuhi kebutuhan pengguna akhir. Dalam hal ini, analisis secara menyeluruh tidak dilakukan, melainkan hanya sebatas memenuhi kebuthan akan fungsi perangkat lunak tersebut.

2.4 Metode *McCall*

2.4.1 McCall

McCall adalah hubungan antara karakteristik kualitas dan metrik, walaupun terdapat kritik bahwa tidak semua metric adalah obyektif. Salah satu aspek yang tidak dipertimbangkan langsung oleh model ini adalah fungsionalitas dari produk perangkat lunak.

Model *McCall* mencoba untuk menjembatani kesenjangan antara pengguna dan pengembang dengan berfokus pada sejumlah faktor kualitas perangkat lunak yang mencerminkan pandangan pengguna dan prioritas pengembang. Gagasan utama dalam model *McCall* adalah untuk menilai relativitas hubungan social antara factor-faktor kualitas eksternal dan kriteria kualitas produk. Model ini dikembangkan oleh angkatan udara Amerika Serikat pada system keputusan elektronik (*Electronic System Decision*), pusat pengembangan

Rome Air (*Rome Air Development Center*) dan *General Electric* (GE), dengan maksud meningkatkan kualitas produk perangkat lunak [5].

2.4.2 Parameter-parameter Penilaian McCall Model

Model ini terdiri dari 11 (sebelas) parameter penilaian dan memiliki tiga perspektif (*focus*) utama yaitu [5]:

1. *Product operation* (sifat-sifat operasional dari *software*), operasi produk yang berhubungan dengan kemampuan produk agar mudah dipahami dan pengoperasian yang efisien, terdiri dari :
   a. *CORRECTNESS*
   b. *RELIABILITY*
   c. *EFFICIENCY*
   d. *INTEGRITY*
   e. *USABILITY*
2. *Product revision* (daya/kemampuan *software* dalam menjalani perubahan), revisi produk berhubungan dengan pemeriksaan kesalahan dan adaptasi *system*, terdiri dari :
   a. *MAINTAINABILITY*
   b. *FLEXIBILITY*
   c. *TESABILITY*
3. *Product transition* (daya/kemampuan adapatasi *software* terhadap lingkungan baru), transisi produk berhubungan dengna proses terdistribusi dan adapatasi *hadware* yang mudah, yang terdiri dari :
   a. PORTABILITY
   b. REUSABILITY
   c. INTEROPERABILITY

Untuk membentuk pengukuran langsung mengenai faktor-faktor kualitas tidaklah mudah. Terdapat beberapa ukuran (metric) yang didefinisikan dan penilaiannya diukur secara objectif. Pengukuran biasanya dalam bentuk *checklist* dengan menggunakan skala 0-10. *McCall* menetapkan beberapa pengukuran yang dapat digunakan, diantaranya:

1. *Auditability*, kemudahan yaitu penyesuaian terhadap standar yang dapat diperiksa.
2. *Accuracy*, ketepatan perhitungan dan control
3. *Communication commonality*, tingkatan dimana interface standar, protocol dan bandwidth digunakan

4. *Completenesss*, tingkatan dimana implekmentasi lengka pdari fungsi yagn dibutuhkan telah tercapai
5. *Conciseness*, kepadatan program dalam jumlah baris kode
6. *Consistency*, penggunaan rancangan dan teknik dokumentasi dalam satu bentuk diseluruh proyek pengembangan *software*
7. *Datacommonality*, penggunaan struktur dan tipe data standar diseluruh program
8. *Error tolerance*, kerusakan yang muncil ketika program menemukan kesalahan/kegagalan
9. *Execution efficiency*, performa run-time suatu program
10. *Expandability*, tingkatan dimana rancangan arsitektural, data atau prosedur dapat dikembangkan
11. *Generality*, lingkup alikasi potensial dari suatu komponen program
12. *Hardware independence*, tingkatan dimana software dipisahkan dari hadware yang mengoperasikannya
13. *Instrumentation*, tingkatan dimana pengawasan program memiliki operasi tersendiri dan mengidentifikasi kesalahan yang terjadi
14. *Modularity*, kemandirian fungsional dari suatu komponen program
15. *Operability*, kemudahan pengoperasian program
16. *Security*, ketersediaan mekanisme yang mengontrol atau menproteksi program dan data
17. *Self-documentation*, tingkatan dimana kode sumber menyediakan dokumentasi yang berarti
18. *Simplicity*, tingkatan dimana program dapat dimengerti tanpa kesulitan
19. *Software system independence*, tingkatan dimana program mandiri terhadap feature Bahasa pemrograman nonstandar, karateristik system operasi, dan batasan-batasan lingkunganlainnya
20. *Traceability*, kemampuan penelusuran ulang kepada kebutuhan mengenai representasi rancangan atau komponen program yang sesungguhnya
21. *Training*, tingkatan dimana *software* membantu menerapkan system oleh user yang baru.

2.4.3 Metode Penilaian McCall

Tahap analisis data ini menggunkan teknik analisis Kuantitatif. Berikut dibawah ini adalah metode penilaian yang dikemukakan oleh McCall sebagai landasan pada pembuatan instrument penelitian [7].

1. Tentukan kriteria yang digunakan untuk mengukur suatu factor.
2. Menentukan bobot (w) dari setiap kriteria (0 <= w <= 1).
3. Menentukan skala kriteria, dimana skala penilaian yang digunakan antara 1 – 5, dimana 1 adalah penilaian minimum dan 5 penilaian maksimum.
4. Memasukkan nilai pada tiap kriteria hasil dari penilaian responden.
5. Rumus McCall

$$Fa = w1c1 + w2c2 + \cdots + wncn$$

Keterangan :

Fa : Nilai total dari faktor a

wi : bobot untuk kriteria i

ci : nilai untuk kriteria i

6. Kemudian penjumlahan total dikalikan 100% dengan ketentuan bobot nilai dalam persen adalah sebagai berikut:

$$\text{Persentase} = \frac{Nilai\ yang\ didapat}{Nilai\ maksimum} \times 100\%$$

Hasil persentase digunakan untuk memberikan jawaban atas kelayakan dari aspek-aspek yang diteliti. Pembagian kategori kualitas ada lima. Skala ini memperhatikan rentang dari bilangan persentase. Nilai maksimal yang diharapkan adalah 100 % dan minimum 0 % . Pembagian rentang kategori kualitas dapat dilihat dibawah ini :

81%-100% = Sangat Baik
61%-80% = Baik
41%-60% = Cukup Baik
21%-40% = Tidak Baik
<21% = Sangat Tidak Baik