

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi bukan merupakan hal yang baru. Yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan pemimpin merencanakan serta mengendalikan operasi yang sudah ada. komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan. Suatu organisasi terdiri atas sejumlah unsur, orang-orang yang mempunyai berbagai peran. [5] Kegiatan atau tugas yang harus diselesaikan, tempat kerja, wewenang, serta hubungan komunikasi yang menikat organisasi tersebut. Sistem informasi merupakan penerapan sistem di dalam organisasi untuk mendukung informasi yang dibutuhkan oleh semua tingkat manajemen. Informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Informasi diperoleh dari sistem informasi. [5]

Sistem informasi adalah suatu sistem yang di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. [5] Secara konsep sistem informasi, komputer tidak harus digunakan di dalam sistem, akan tetapi kenyataannya tidaklah mungkin suatu sistem yang kompleks dapat melibatkan elemen non-komputer dan elemen komputer. Terdapat dua kelompok pendekatan di dalam pendefinisian konsep sistem informasi, yaitu kelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. [6] Pendekatan yang menekankan pada prosedur mendefinisikan sistem informasi sebagai suatu jaringan kerja sesuai prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem informasi yang lebih menekankan pada elemen atau komponen mendefinisikan sistem informasi sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Kedua kelompok konsep sistem informasi ini adalah benar dan tidak bertentangan. Yang berbeda adalah cara pendekatannya. [6] Secara sederhana sistem dan informasi dapat diartikan sebagai suatu kumpulan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling

bergantung satu sama lain dan terpadu. Teori konsep sistem dan informasi secara umum pertama kali diuraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem informasi. [6]

Sistem informasi dan teknologi kini telah menjadi komponen yang sangat penting bagi keberhasilan bisnis dan organisasi. Teknologi informasi, termasuk sistem informasi berbasis internet, memainkan peranan penting dan makin luas dalam bisnis. Teknologi informasi dapat membantu segala jenis bisnis, meningkatkan efisiensi dan efektivitas proses bisnis mereka, pengambilan keputusan manajerial dan kerja sama kelompok kerja sehingga dapat memperkuat posisi kompetitif mereka dalam pasar yang cepat sekali berubah. Hal ini berlaku ketika teknologi informasi digunakan untuk mendukung tim pengembangan produk, proses dukungan untuk pelanggan, transaksi *e-commerce* atau dalam aktivitas bisnis lainnya. Teknologi dan sistem informasi berbasis internet dalam waktu singkat menjadi bahan yang dibutuhkan untuk keberhasilan bisnis di lingkungan global yang dinamis saat ini. [6]

Sistem informasi memiliki beberapa komponen-komponen diantaranya : [7]

a. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Yang dimaksudkan dengan input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.

b. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok keluaran (*output block*)

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

d. Blok teknologi (*technology block*)

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

e. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*database management system*).

f. Blok kendali (*control block*)

Banyak hal dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan pada sistem itu sendiri, ketidak-efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi.

Sistem informasi memiliki beberapa manfaat diantaranya: [6]

1. Data yang terpusat

Sistem informasi menjadikan data dan informasi terkumpul secara terpusat pada satu tempat, yaitu *database*. *Database* berada di komputer *server* sistem informasi, yang memuat satu atau beberapa buah tabel sesuai keperluan.

2. Kemudahan di dalam mengakses informasi

Sistem informasi menjadikan anda begitu mudah mengakses dan menikmati sajian informasi yang diberikan. Anda bisa menyimak di komputer *desktop*, laptop dan genggam melalui *smartphone*. Selain itu, kita dapat turut serta di dalamnya (mengirim berita, memberi komentar, opini dan lain-lain secara cepat).

3. Efisiensi waktu

Dengan adanya sistem informasi, maka kemudahan seperti yang dijelaskan di bagian atas mempengaruhi efisiensi waktu anda. Cukup dengan terkoneksi ke jaringan/ *server* sistem

informasi melalui *gadget*/ komputer, anda dapat menyimak semua informasi yang disajikan saat itu juga.

4. Cakupan dan penyebaran informasi menjadi lebih luas dan cepat

Dengan adanya sistem informasi, cakupan informasi yang disajikan tidak lagi hanya untuk perorangan atau beberapa orang, namun dapat secara umum ke siapapun yang mengakses sistem informasi. Hal ini akan menjadikan informasi dapat lebih cepat tersebar luas, sehingga informasi tersebut dapat makin bermanfaat.

5. Memudahkan proses bisnis dan pekerjaan

Melalui sebuah sistem informasi, pekerjaan yang berat dan dilakukan secara manual oleh petugas (*operator*) dapat dikerjakan dengan lebih mudah, otomatis, dan lebih hemat waktu, dengan hasil yang lebih baik. Misalkan pencarian data, *input* data, dan pengolahan data.

6. Biaya murah untuk akses dan penyediaan informasi

Sistem informasi menawarkan biaya yang murah untuk mengakses informasi. Cukup dengan biaya sewa internet saja, anda sudah dapat meraup sebanyak mungkin informasi berharga yang anda perlukan. Di sisi penyedia layanan informasi, biaya lebih murah untuk menyajikan informasi kedalam bentuk sistem informasi (misalkan dalam bentuk *website*/ sistem informasi berbasis *web*), dibandingkan mencetaknya secara massal kedalam bentuk koran/ selebaran maupun tayangan di televisi/ radio.

7. Menyimpan data lebih banyak dengan ruang yang lebih kecil

Sistem informasi menghemat ruang penyimpanan data dan informasi anda. Sebuah sistem informasi hanya memerlukan minimal sebuah komputer biasa saja (dengan kelengkapan komponen didalamnya). Tentu saja, ini hanya membutuhkan sebuah ruangan yang kecil dan pasokan listrik yang tidak relatif besar. Bandingkan dengan sistem yang lama dalam bentuk berkas (kertas), dimana diperlukan banyak rak/ lemari dan ruang yang relatif lebih luas.

8. Solusi komunikasi yang murah, hemat dan andal

Melalui sebuah sistem informasi yang telah dilengkapi dengan sarana komunikasi *online*, maka sistem informasi dapat memenuhi kebutuhan akan layanan komunikasi yang murah, hemat, andal, dan bermanfaat bagi *internal* organisasi tempat sistem informasi tersebut berada.

9. Penyimpanan data dapat lebih berkembang sesuai kebutuhan

Sistem informasi dengan media penyimpanan berupa basis data (*database*), memiliki kemampuan untuk dapat berkembang jauh lebih besar lagi sesuai keperluan (*extensibility, scalable*).

2.2 Rekayasa Perangkat Lunak

Suatu Perangkat Lunak menjadi kebutuhan manusia dengan berbagai bagian disiplin ilmu yang dibidangi setiap tenaga profesionalnya, menjadi bagian penting yang melatarbelakangi tumbuhnya perkembangan perangkat lunak dengan berbagai krisis perangkat lunak menurut berbagai sisi pandang konsumen, manajer dan pengembang/praktisi. Rekayasa Perangkat lunak pada dasarnya merupakan aplikasi dari suatu pendekatan yang sistematis, disiplin, dan dapat diukur pada pengembangan, operasi, dan perawatan perangkat lunak. [7] Perangkat lunak merupakan program-program komputer dan dokumentasi yang berkaitan dan Perangkat Lunak (*Software*) adalah *source code* pada suatu program atau sistem. Perangkat lunak tidak hanya dokumentasi terhadap *source code* tapi juga dokumentasi terhadap sesuatu yang dibutuhkan selama pengembangan, instalasi, penggunaan dan pemeliharaan sebuah sistem.

Engineering atau rekayasa adalah aplikasi terhadap sistematis yang berdasar atas ilmu pengetahuan dan matematis serta aplikasi tentang produksi terhadap struktur, mesin, produk, proses atau sistem. [7]

Kategori Perangkat lunak secara umum dapat dikelompokkan sebagai berikut : [7]

1. Perangkat Lunak Sistem adalah sekumpulan program yang ditulis untuk melayani program-program yang lain. Seperti kompilator, editor dan utilitas pengatur file.
2. Perangkat Lunak *Real-Time*, adalah program-program yang memonitor, menganalisa, dan mengontrol kejadian dunia nyata pada saat terjadinya (*real time event*)
3. Perangkat Lunak Bisnis berfungsi untuk memroses informasi bisnis seperti *payroll, inventory*.
4. Perangkat Lunak Teknik dan Ilmu Pengetahuan, ditandai dengan penggunaan algoritma *number crunching*.
5. *Embedded Software*, produk yang ada dalam *read-only memory* dan dipakai untuk mengontrol hasil dan sistem untuk keperluan konsumen dan pasar industry
6. Perangkat Lunak Komputer Personal, sesuai kebutuhan personal seperti pengolah kata, angka dan manajemen database.

7. Perangkat Lunak Kecerdasan Buatan, menggunakan algoritma non-numeris untuk memecahkan masalah kompleks yang tidak sesuai untuk perhitungan atau analisis secara langsung/

2.2.1 Karakteristik Perangkat Lunak

Atribut Perangkat Lunak seharusnya memberikan pengguna kebutuhan fungsionalitas dan unjuk kerja yang dapat di rawat, berguna. Perangkat Lunak mempunyai Karakteristik sebagai berikut : [8]

1. *Maintanability* adalah Perangkat Lunak harus dapat memenuhi perubahan kebutuhan
2. *Dependability* adalah Perangkat Lunak harus dapat dipercaya.
3. *Efisiensi* adalah Perangkat Lunak harus efisien dalam penggunaan resource.
4. *Usability* adalah Perangkat Lunak harus dapat digunakan sesuai dengan yang direncanakan.

Proses perangkat lunak merupakan serangkaian kegiatan atau aktivitas yang hasil-hasil relevannya menghasilkan suatu perangkat lunak. Kegiatan ini biasanya dilakukan oleh perekayasa perangkat lunak

Aktifitas umum dalam semua proses perangkat lunak terdiri dari: [8]

1. Membuat Spesifikasi Perangkat Lunak yaitu kegiatan ini menggambarkan fungsionalitas suatu perangkat lunak dan batasan-batasan kemampuan operasinya yang telah didefinisikan.
2. Pengembangan Perangkat Lunak yaitu mengetahui spesifikasi kebutuhan dari perangkat lunak tersebut, maka perangkat lunak yang telah memenuhi spesifikasi tersebut harus segera diproduksi.
3. Validasi Perangkat Lunak yaitu perangkat lunak harus sering divalidasi, mengapa? Karena untuk menjamin bahwa perangkat lunak tersebut melakukan apa yang seharusnya diinginkan/dibutuhkan oleh pelanggan.
4. Evolusi Perangkat Lunak yaitu perangkat lunak juga haruslah dapat berkembang untuk memenuhi kebutuhan pelanggan dimasa yang akan datang.

2.3 Pengujian Perangkat Lunak

Pada proses rekayasa perangkat lunak, pelaku rekayasa perangkat lunak mula-mula berusaha untuk membangun perangkat lunak mulai dari konsep abstrak sampai kepada tahap implementasi yang dapat dilihat, baru kemudian dilakukan pengujian. Pada pengujian perangkat lunak, pelaku rekayasa perangkat lunak menciptakan sekumpulan kasus uji untuk diujikan kepada

perangkat lunak. Proses ini lebih terkesan berusaha untuk “membongkar” perangkat lunak yang sudah dibangun. Proses pengujian merupakan tahapan dalam rekayasa di mana secara fisik terlihat lebih banyak sisi deskruktifnya dibandingkan sisi konstruktifnya karena tujuannya adalah untuk menemukan kesalahan pada perangkat lunak. [9]

Berikut ini adalah tahapan pengujian perangkat lunak sebagai berikut [9]:

1. Pengujian unit (*unit testing*) menguji komponen perangkat lunak komponen atau modul.
2. Pengujian Integrasi (*integration testing*) menjelaskan kecacatan dalam antarmuka dan interaksi antarkomponen terpadu (modul).
3. Pengujian sistem (*system testing*) menguji sistem terpadu secara penuh untuk memastikan bahwa *system* telah memenuhi persyaratan.
4. Pengujian sistem integrasi (*system integration testing*) memverifikasi sistem terpadu untuk semua sistem eksternal atau pihak ketiga yang telah ditetapkan di dalam persyaratan *system*.

Sasaran pengujian perangkat lunak antara lain: [9]

1. Pengujian adalah proses mengeksekusi program dengan tujuan khusus untuk menemukan kerusakan
2. Kasus uji yang baik adalah yang memiliki tingkat kemungkinan tinggi untuk menemukan kerusakan yang belum ditemukan
3. Pengujian dikatakan berhasil jika berhasil menemukan kerusakan yang belum ditemukan.

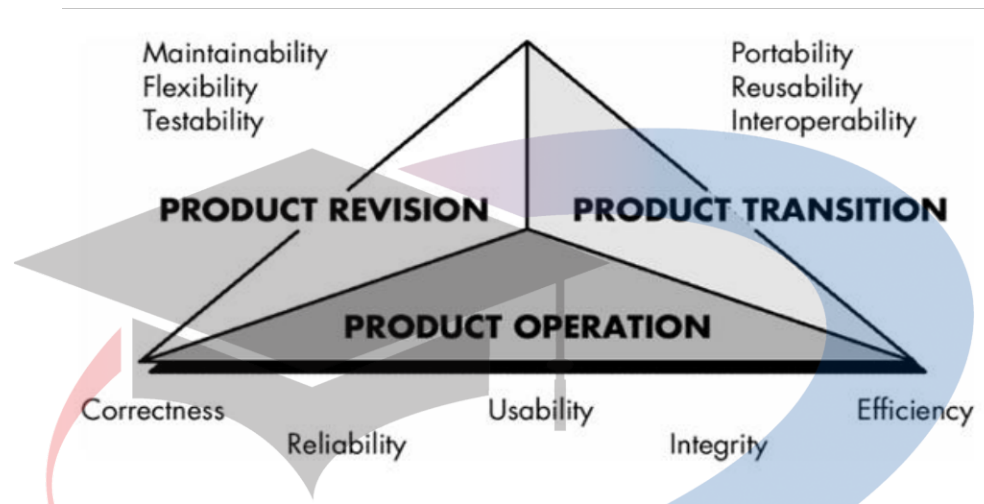
Sasaran di atas sekaligus mengimplikasikan adanya perubahan cara pandang di mana sebelumnya dikatakan bahwa pengujian yang berhasil adalah pengujian yang tidak menemukan kesalahan. Jika pengujian sukses dilakukan, maka akan ditemukan kesalahan dalam perangkat lunak. Sebagai keuntungan tambahan, pengujian menunjukkan bahwa fungsi perangkat lunak bekerja sesuai spesifikasi dan bahwa persyaratan kinerja telah dipenuhi. Data yang dikumpulkan pada saat pengujian dilakukan memberikan indikasi yang baik mengenai realibilitas perangkat lunak dan beberapa indikasi dari kualitas keseluruhan perangkat lunak. Akan tetapi, pengujian tidak dapat memperlihatkan bahwa perangkat lunak yang diuji tidak memiliki cacat. [9]

Serangkaian prinsip pengujian perangkat lunak antara lain: [9]

1. Semua pengujian harus dapat ditelusuri sampai ke kebutuhan pelanggan. Sasaran pengujian perangkat lunak adalah untuk menemukan kesalahan. Hal ini memenuhi kriteria bahwa cacat yang paling fatal dilihat dari sisi pandang pelanggan adalah cacat yang mengakibatkan program gagal memenuhi persyaratannya
2. Pengujian harus direncanakan lama sebelum pengujian dimulai. Perencanaan pengujian dapat dimulai segera setelah model kebutuhan dilengkapi. Definisi detail kasus uji dapat dibuat segera setelah model perancangan dibuat. Dengan demikian, semua pengujian dapat direncanakan dan dirancang sebelum semua kode dibangkitkan
3. Prinsip *Pareto* berlaku untuk pengujian perangkat lunak. Prinsip *Pareto* mengimplikasikan bahwa 80% dari semua kesalahan yang ditemukan selama pengujian akan dapat ditelusuri sampai 20% dari modul program. Permasalahannya adalah bagaimana mengisolasi modul yang dicurigai dan mengujinya dengan teliti
4. Pengujian harus mulai “dari yang kecil” dan berkembang ke pengujian “yang besar”. Pengujian pertama kali dilakukan fokus pada modul individual perangkat lunak, kemudian pengujian mengubah fokus menjadi menemukan kesalahan pada cluster modul yang terintegrasi, dan akhirnya pada sistem secara keseluruhan
5. Tidak mungkin melakukan pengujian yang mendalam. Jumlah jalur permutasi untuk program yang berukuran menengah sangat besar. Karena itulah, tidak mungkin untuk mengeksekusi setiap kombinasi jalur skema pengujian. Tetapi dimungkinkan untuk secara memadai mencakup logika program dan memastikan bahwa semua kondisi dalam rancangan prosedural telah diuji
6. Agar memperoleh pengujian yang paling efektif, pengujian harus dilakukan oleh pihak ketiga yang independen. Maksudnya, agar pengujian memiliki tingkat kemungkinan yang tinggi untuk menemukan kesalahan, maka pelaku rekayasa perangkat lunak yang membuat sistem tersebut bukanlah orang yang paling tepat untuk melakukan semua pengujian bagi perangkat lunak

2.4 Metode McCall

Beberapa model faktor kualitas perangkat lunak dan kategorisasinya sudah diusulkan selama bertahun-tahun. Model klasik dari faktor kualitas perangkat lunak dikemukakan oleh McCall yang terdiri dari 11 faktor. Penjelasan dari masing-masing faktor kualitas tersebut adalah sebagai berikut [9]:



Gambar 1.1 Metode McCall

1. *Correctness* adalah sebuah perangkat lunak dapat dikatakan benar jika memenuhi persyaratan sebagai berikut:
 - a. Menghasilkan keluaran yang benar untuk setiap kemungkinan masukan oleh pengguna.
 - b. Melakukan proses yang seharusnya (tidak kurang dan tidak berlebihan).
 - c. Secara formal harus bisa dibuktikan secara matematis.
2. *Reliability* adalah Sudut pandang reliabilitas pada poin ini lebih menekankan pada kemungkinan dari *failure-free* suatu operasi perangkat lunak terhadap periode waktu tertentu di dalam lingkungan tertentu. Software *reliability* bukan fungsi langsung terhadap waktu.
3. *Efficiency* adalah sebuah perangkat lunak yaitu penggunaan sumber daya seperti waktu pemrosesan processor (*eksekusi*), pemakaian media penyimpanan (*memori, space, bandwidth*) serta berkaitan dengan hubungan antara kinerja perangkat lunak dan jumlah sumber daya yang digunakan.

4. *Integrity* adalah Integritas perangkat lunak pada model *McCall* lebih menekankan kepada keamanan sebuah perangkat lunak. Pihak *developer* harus mampu melihat kebutuhan akan hak akses perangkat lunak tersebut pada setiap penggunaanya.
5. *Usability* Faktor ini melihat dari kemudahan perangkat lunak untuk digunakan dan dipelajari. *Usability* mempunyai unsur akademis seperti *psikologis*, *ergonomi*, dan *human factors*.
6. *Maintainability* adalah kemudahan dari perangkat lunak untuk dipelihara, seperti:
 - a. Memperbaiki kerusakan
 - b. Menemukan kebutuhan baru
 - c. Membuat pemeliharaan selanjutnya lebih mudah
 - d. Mengatasi lingkungan yang berubah. Sebuah perangkat lunak dikatakan dapat dipelihara jika koreksi dari minor bugs memerlukan usaha yang kecil.
7. *Flexibility* dua faktor fleksibilitas perangkat lunak, yaitu: kemudahan yang didalam membuat perubahan yang dibutuhkan akibat perubahan lingkungan serta kemampuan melakukan modifikasi kode untuk memfasilitasi perubahan yang telah ditentukan.
8. *Testability* yaitu kemampuan perangkat lunak untuk diuji. Selain itu *testability* adalah derajat yang dimiliki sebuah sistem untuk memfasilitasi kriteria pengujian dan perfomansi dari pengujian tersebut untuk mengukur sejauh mana kriteria tersebut dipenuhi.
9. *Portability* Perangkat lunak dikatakan portabel jika biaya untuk memindahkannya (transport dan adaptasi) ke lingkungan yang baru lebih kecil jika dibandingkan dengan biaya untuk membangun perangkat lunak tersebut dari awal.
10. *Reusability* adalah properti dari perangkat lunak yang memungkinkan perangkat lunak atau modul-modulnya digunakan kembali untuk sistem lain. Suatu perangkat lunak dikatakan *reusable* yang baik jika modul-modulnya dapat digunakan kembali untuk aplikasi lainnya.
11. *Interoperability* adalah kemampuan suatu perangkat lunak untuk bekerja dengan perangkat lunak lainnya tanpa mengalami kesulitan.

Pada dasarnya *McCall* menitik beratkan faktor-faktor tersebut menjadi 3 (tiga) aspek penting, yaitu yang berhubungan dengan : [9]

1. Sifat-sifat operasional dari software (*Product Operation*).
2. Kemampuan software dalam menjalani perubahan (*Product Revision*).
3. Daya adaptasi atau penyesuaian software terhadap lingkungan baru (*Product Transition*).

Teknik Pengukuran Dalam menghitung kualitas menggunakan metode *McCall* menggunakan Persamaan :

$$f_q = c_1 * m_1 + c_2 * m_2 + c_3 * m_3 + \dots + c_n * m_n$$

dimana :

F_q = Faktor *software quality*

c_1 = Bobot yang bergantung pada produk dan kepentingan

m = Metric yang mempengaruhi faktor *software quality*

Tahapan yang harus kita tempuh dalam pengukuran adalah sebagai berikut:

Tahap 1: Tentukan kriteria yang digunakan untuk mengukur suatu faktor

Tahap 2: Tentukan bobot (w) dari setiap kriteria (biasanya $0 \leq w \leq 1$)

Tahap 3: Tentukan skala dari nilai kriteria (misalnya, $0 \leq \text{nilai kriteria} \leq 10$)

Tahap 4: Berikan nilai pada tiap kriteria

Tahap 5: Hitung nilai total dengan rumus

$$f_a = w_1c_1 + w_2c_2 + \dots + w_nc_n$$

Adapun metrik yang dipakai dalam skema pengukuran di atas adalah sebagai berikut: [9]

1. *Auditability* adalah kemudahan untuk memeriksa apakah software memenuhi standard atau tidak.
2. *Accuracy* adalah ketelitian dari komputasi dan control
3. *Communication Commonality* adalah sejauh mana interface, protokol, dan bandwidth digunakan.
4. *Completeness* adalah sejauh mana implementasi penuh dari fungsifungsi yang diperlukan telah tercapai.
5. *Conciseness* adalah keringkasan program dalam ukuran LOC (line of commands)
6. *Consistency* adalah derajat penggunaan teknik-teknik desain dan dokumentasi yang seragam pada seluruh proyek pengembangan software.

7. *Data Commonality* adalah derajat penggunaan tipe dan struktur data baku pada seluruh program.
8. *Error Tolerance* adalah kerusakan yang terjadi apabila program mengalami error.
9. *Execution Efficiency* adalah kinerja run-time dari program.
10. *Expandability* adalah sejauh mana desain prosedur, data, atau arsitektur dapat diperluas.
11. *Generality* adalah luasnya kemungkinan aplikasi dari komponen-komponen program.
12. *Hardware Independence* adalah sejauh mana software tidak bergantung pada kekhususan dari hardware tempat software itu beroperasi.
13. *Instrumentation* adalah sejauh mana program memonitor operasi dirinya sendiri dan mengidentifikasi error yang terjadi.
14. *Modularity* adalah functional independence dari komponen-komponen program.
15. *Operability* adalah kemudahan mengoperasikan program.
16. *Security* adalah ketersediaan mekanisme untuk mengontrol dan melindungi program dan data terhadap akses dari pihak yang tidak berhak.
17. *Self-Dokumentation* adalah sejauh mana source-code memberikan dokumentasi yang berarti.
18. *Simplicity* adalah Kemudahan suatu program untuk dimengerti.
19. *Traceability* adalah kemudahan merujuk balik implementasi atau komponen program ke kebutuhan pengguna software.
20. *Training* adalah sejauh mana software membantu pemakaian baru untuk menggunakan sistem.

Dari beberapa faktor kualitas menurut taksonomi McCall, untuk menentukan kualitas perangkat lunak cukup dengan lima faktor. Lima faktor untuk menentukan kualitas perangkat lunak tersebut adalah :

1. Ketepatan (*Correctness*)
2. Keandalan (*Reliability*)
3. Efisiensi (*Efficiency*)
4. Kegunaan (*Usability*)
5. Dapat dijalankan hardware dan software yang ada (*Compability*).

Pengujian perangkat lunak dapat dinyatakan sebagai proses *validasi* dan *verifikasi* bahwa sebuah program / aplikasi / produk adalah sebagai berikut : [10]

1. Memenuhi kebutuhan (*requirement*) yang mendasari perancangan dan pengembangan perangkat lunak tersebut.
2. Berjalan sesuai dengan yang diharapkan.
3. Dapat diterapkan menggunakan karakteristik yang sama.
4. Memenuhi kebutuhan semua pihak yang berkepentingan.

Black-Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Contoh pengujian perangkat lunak dengan teknik *Black Box* Pemeliharaan data untuk aplikasi bank yang sudah diotomatisasikan. Pemakai dapat memutar nomor telepon bank dengan menggunakan mikrokomputer yang terhubung dengan password yang telah ditentukan dan diikuti dengan perintah-perintah.

Data yang diterima adalah :

Kode area : kosong atau 3 digit.
 Prefix : 3 digit atau tidak diawali 0 atau 1.
 Password : 6 digit alfanumerik.
 Perintah : *check*, *deposit*, dll.

Selanjutnya kondisi input digabungkan dengan masing-masing data elemen dapat ditentukan sebagai berikut:

Kode area : kondisi input, Boolean - kode area mungkin ada atau tidak kondisi input, range-nilai ditentukan antara 200 dan 999.

Prefix : kondisi input range > 200 atau tidak diawali 0 atau 1.

Suffix : kondisi input nilai 4 digit.

Password : kondisi input boolean – *password* mungkin diperlukan atau tidak kondisi input nilai dengan 6 karakter string.

Perintah : kondisi input set berisi perintah-perintah yang telah didefinisikan.

Terdapat cukup banyak pendekatan yang dilakukan untuk melakukan *testing*. Salah satu definisi *testing* adalah “sebuah proses yang melakukan pertanyaan terhadap sebuah produk untuk dinilai”, di mana “pertanyaan” merupakan segala sesuatu yang diberikan kepada produk sebagai pengujian. Beberapa tahapan testing yang umum dilalui oleh aplikasi adalah sebagai berikut : [10]

1. *Unit/Component Testing*

Terbagi atas *testing* terhadap unit dan komponen. *Unit testing* merupakan proses *testing*, di mana melakukan *testing* pada bagian *basic* dari kode program. Contohnya adalah memeriksa kode program pada *event*, *procedure*, dan *function*. *Unit testing* meyakinkan bahwa masing-masing unit tersebut berjalan sebagaimana mestinya. Pada unit *testing*, memeriksa bagian kode program secara terpisah dari bagian yang lain. Dapat langsung melakukan unit *testing* setiap kali sebuah kode unit (*event*, *procedure*, *function*) selesai dibuat. Lalu memeriksa kode unit dengan menjalankannya baris per baris untuk memastikan bahwa proses yang dilakukan berjalan sebagaimana yang yang diinginkan.

2. *Integration Testing*

Setelah melakukan *Unit/Component Testing*, langkah berikutnya adalah memeriksa bagaimana unit-unit tersebut bekerja sebagai suatu kombinasi, bukan lagi sebagai suatu unit yang individual. Sebagai contoh, memiliki sebuah proses yang dikerjakan oleh dua *function*, di mana satu *function* menggunakan hasil output dari *function* yang lainnya. Kedua *function* ini telah berjalan dengan baik secara individu pada unit *testing*. Pada tahap *Integration Testing*, memeriksa hasil dari interaksi kedua *function* tersebut, apakah bekerja sesuai dengan hasil yang diharapkan. Dan harus memastikan bahwa seluruh kondisi yang mungkin terjadi dari hasil interaksi antar unit tersebut menghasilkan output yang diharapkan.

3. *System Testing*

Mencakup *testing* aplikasi yang telah selesai di-*develop*. Karena itu, aplikasi harus terlihat dan berfungsi sebagaimana mestinya terhadap *end-user* atau pengguna akhir. Untuk itu, *testing* dilakukan dengan menggunakan data yang menggambarkan data yang digunakan oleh pengguna sesungguhnya terhadap aplikasi. Jika aplikasi di-*develop* untuk lingkungan yang besar, lalu dapat melakukan *testing* pada dua komputer yang berbeda. Komputer yang digunakan sebagai komputer *testing* harus terlebih dahulu dikonfigurasi hanya dengan:

- a. *Operating system* yang dibutuhkan.
- b. *Driver* yang diperlukan oleh aplikasi.
- c. Kemudian tes aplikasi.

Dengan menggunakan konfigurasi yang paling minimal dan sederhana, maka dapat membantu untuk memastikan bahwa permasalahan yang timbul selama *testing* berlangsung adalah merupakan kesalahan aplikasi, dan bukan kesalahan yang berasal dari aplikasi atau *software* lain.

4. *Acceptance Testing*

Seperti *Integration testing*, *Acceptance testing* juga meliputi *testing* keseluruhan aplikasi. Perbedaannya terletak pada siapa yang melakukan *testing*. Pada tahap ini, *end-user* yang terpilih melakukan *testing* terhadap fungsi-fungsi aplikasi dan melaporkan permasalahan yang ditemukan. *Testing* yang dilakukan merupakan simulasi penggunaan nyata dari aplikasi pada lingkungan yang sebenarnya. Proses ini merupakan salah satu tahap *final* sebelum pengguna menyetujui dan menerima penerapan sistem aplikasi yang baru. Karena itu pada tahap ini sudah tidak difokuskan untuk mengangkat permasalahan kecil seperti kesalahan pengetikan.

5. *Regression Testing*

Merupakan bagian penting dari masing-masing tahap proses *testing*. *Regression testing* mencakup pengujian ulang terhadap unit, komponen, proses, atau keseluruhan aplikasi setelah perbaikan suatu kesalahan dilakukan. *Regression testing* memastikan permasalahan yang terjadi telah ditanggulangi, dan tidak terdapat permasalahan baru yang timbul sebagai efek perbaikan tersebut. Selain itu, tahap ini tidak hanya berguna untuk melakukan pengujian aplikasi, tetapi dapat juga digunakan untuk melakukan pemantauan kualitas dari output yang dihasilkan. Sebagai contoh, *Regression testing* memantau ukuran file, waktu yang dibutuhkan untuk melakukan suatu tes, waktu yang dibutuhkan untuk melakukan kompilasi, dan lain sebagainya. [10]

Contoh kriteria penilaian

Dengan menentukan nilai rata-rata pada tiap kriteria dan bobot yang bergantung pada kepentingan maka hasil penelitian kualitas perangkat lunak yang diperoleh dari 18 responden dapat dilihat pada tabel hasil penilaian kualitas perangkat lunak berikut ini:

No	Indikator	Keterangan	Bobot	Nilai Kriteria
		a. Completeness (Kelengkapan)		
1	<i>Correctness</i> (Ketepatan)	1. Aplikasi ini sudah mampu melakukan proses pengolahan data	0,4	4,5

	2. Fitur-fitur yang terdapat pada Sistem <i>Inventori</i> sudah berfungsi semua.	0,4	4,4
	<i>b. Consistency (Konsistensi)</i>		
	3. Fitur dan desain tabel di setiap halaman sama	0,2	3,7
	4. Fitur desain <i>form</i> dan tombol di setiap halaman sama	0,2	4,5
	5. Pengelolaan data di setiap form sama	0,2	4,3
	6. Bahasa yang digunakan sudah konsisten pada setiap halamannya	0,3	2,5
	7. Bentuk dan struktur pelaporan pengolahan semua data sama	0,2	3,7
	<i>c. Treacebility (Lacak)</i>		
	8. Pengguna dapat melacak kapan waktu (tanggal dan jam) pengelolaan barang	0,4	4,5
	<i>d. Communicativeness (Komunikatif)</i>		
2	<i>Usabilty</i> (kegunaan)		
	9. Bahasa yang digunakan mudah dipahami	0,4	2,5
	10. Tulisan dari setiap halaman dapat terbaca dengan jelas	0,4	4,4

		11. Fungsi dari setiap tombol jelas	0,4	3,7
		<i>e. Operability (operabilitas)</i>		
		12. Pilihan menu dan tombol pada sistem mudah digunakan	0,4	3,1
		13. Pengguna mudah mengerti dengan sistem pengkodean yang ada	0,4	2,5
		<i>f. Training (pelatihan)</i>		
		14. Pengguna baru dapat dengan mudah menggunakan semua fitur pada aplikasi Tokopedia	0,4	3
2	<i>Usability (kegunaan)</i>	15. Ada layanan petunjuk (<i>help</i>) yang disediakan oleh sistem guna membantu pengguna baru dalam menggunakan aplikasi Tokopedia.	0,4	2,5
		<i>g. Security (Keamanan)</i>		
		16. Proses Login dapat berjalan dengan benar dan sesuai dengan harapan pengguna	0,3	3,7
3	<i>Integrity (Integritas)</i>	17. Aplikasi ini dapat mengontrol akses pengguna dengan membatasi hak akses	0,4	2,5
		<i>h. Accuracy (Akurasi)</i>		
4	<i>Reliability (Kehandalan)</i>	18. Aplikasi mudah memasukkan input yang diperlukan oleh sistem.	0,4	4,5

19. Aplikasi ini dapat menampilkan data yang tepat dengan kata kunci yang dicari 0,4 4,3

20. Aplikasi ini memberikan data dan informasi yang sesuai dengan kebutuhan pengguna secara tepat 0,3 3,7

21. Informasi dari aplikasi ini akurat dan bebas dari kesalahan 0,4 4,5

22. Pengguna dapat memperoleh informasi yang dibutuhkan dalam waktu yang tepat 0,4 2,9

23. Output dari aplikasi ini disajikan dalam bentuk yang tepat sehingga memudahkan pemahaman pengguna 0,4 2,5

4 *Reliability*
(Kehandalan) i. *Error tolerancy* (toleransi kesalahan)

24. Bagaimana jika akses ke aplikasi dipergunakan oleh pihak yang tidak berhak untuk menggunakannya 0,4 3,7

j. *Simplicity*

25. Informasi yang ada pada aplikasi ini mudah dipahami tanpa ada kesulitan. 0,3 4,5

		26. Menu-menu yang ada pada aplikasi ini dapat mudah dipahami tanpa ada kesulitan	2,5	0,3
5	<i>Efficiency</i> (Efisiensi)	k. <i>Execution efficiency</i> (kemudahan eksekusi)		
		27. Apakah menu layanan fungsi dan datanya sudah sesuai kebutuhan	0,3	3,5
		28. Apakah metode pembayaran yang tersedia sudah memudahkan pengguna	0,3	2,3
		29. Fungsi dari konten pemesanan apakah sudah memudahkan pemahaman pengguna	0,3	2,4

Table 2.1 Hasil Penilaian

Langkah terakhir setelah nilai pembobotan dan nilai kriteria ditentukan yaitu menentukan nilai totalnya F_a berdasarkan pada faktor kualitas yang ada pada *McCall*.

Berikut rumus yang digunakan pada teknik *McCall*:

$$F_a = w_1c_1 + w_2c_2 + \dots + w_nc_n$$

Dimana :

F_a : Nilai total dari faktor a

W : bobot yang bergantung pada produk dan kepentingan

C : adalah metric yang mempengaruhi faktor *software quality* (nilai rata-rata)

Perhitungan masing-masing faktor kualitas dilakukan berdasarkan kriteria yang telah ditentukan adalah sebagai berikut:

a. *Correctness*

$$\begin{aligned} \text{completeness} &= (w_1c_1 + w_2c_2) \\ &= (0,4 \times 4,5) + (0,4 \times 4,4) \\ &= 1,8 + 1,76 = 3,56 \end{aligned}$$

$$\begin{aligned} \text{Consistency} &= (w_3c_3 + w_4c_4 + w_5c_5 + w_6c_6 + w_7c_7) \\ &= (0,2 \times 3,7) + (0,2 \times 4,5) + (0,3 \times 4,3) + (0,2 \times 2,5) + (0,3 \times 3,7) \\ &= 0,74 + 0,9 + 1,29 + 0,5 + 1,11 \\ &= 4,54 \end{aligned}$$

$$\begin{aligned} \text{Treacebility} &= (w_8c_8) \\ &= 0,4 \times 4,5 \\ &= 1,8 \end{aligned}$$

Jadi nilai rata F1 diselesaikan dengan cara berikut:

$$\begin{aligned} fa_1 &= \frac{\text{completeness} + \text{consistency} + \text{treacebility}}{3} \\ &= \frac{(3,56 + 4,54 + 1,8)}{3} \\ &= \frac{9,9}{3} \\ &= 3,3 \end{aligned}$$

Dari hasil yang diperoleh oleh perhitungan kemudian nilai *quality factor* diubah dalam bentuk persentase menggunakan persamaan:

$$\begin{aligned} \text{Persentase} &= \frac{\text{Nilai yang didapat}}{\text{Nilai maksimum}} \times 100\% \\ &= \frac{3,3}{5} \times 100\% \end{aligned}$$

$$= 66\%$$

b. *Usability*

$$\begin{aligned} \text{communicativeness} &= (w_1c_1 + w_2c_2 + w_3c_3) \\ &= (0,4 \times 2,5 + 0,4 \times 4,4 + 0,4 \times 3,7) \\ &= 1 + 1,76 + 1,48 = 4,24 \end{aligned}$$

$$\begin{aligned} \text{Operability} &= w_3c_3 + w_4c_4 \\ &= (0,4 \times 3,1 + 0,4 \times 2,5) \\ &= 1,24 + 1 \\ &= 2,24 \end{aligned}$$

$$\begin{aligned} \text{Training} &= w_5c_5 + w_6c_6 \\ &= (0,4 \times 3 + 0,4 \times 2,5) = 2,2 \end{aligned}$$

$$\text{Fa2} = \frac{\text{communicativeness} + \text{Operability} + \text{Training}}{3}$$

$$= \frac{4,24 + 2,24 + 2,2}{3}$$

$$= \frac{8,68}{3}$$

$$= 2,9$$

Dari hasil yang diperoleh dari perhitungan kemudian kemudian nilai kualitas factor diubah dalam bentuk persentase menggunakan persamaan :

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

$$= \frac{2,9}{5} \times 100\%$$

$$= 58\%$$

c. *Integrity*

$$\begin{aligned}
 \text{Security Fa3} &= (w1c1 + w2c2) \\
 &= (0,3 \times 3,7) + (0,4 \times 2,5) \\
 &= 2,11
 \end{aligned}$$

Dari hasil yang diperoleh dari perhitungan kemudian nilai kualitas faktor diubah dalam bentuk persentase menggunakan persamaan

$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Nilai yang didapat}}{\text{Nilai maksimum}} \times 100\% \\
 &= \frac{2,11}{5} \times 100\% \\
 &= 42,2\%
 \end{aligned}$$

d. *Reliability*

$$\begin{aligned}
 \text{Accuracy} &= (w1c1 + w2c2 + w3c3 + w4c4 + w5c5 + w6c6) \\
 &= (0,4 \times 4,5 + 0,4 \times 4,3 + 0,3 \times 3,7 + 0,4 \times 4,5 + 0,4 \times 2,9 + 0,4 \times 2,5) \\
 &= 8,59
 \end{aligned}$$

$$\begin{aligned}
 \text{Error tolerancy} &= (w7c7) \\
 &= (0,4 \times 3,7) \\
 &= 1,48
 \end{aligned}$$

$$\begin{aligned}
 \text{Simplicity} &= (w8c8 + w9c9) \\
 &= (0,3 \times 4,5 + 0,3 \times 2,5) = 2,1
 \end{aligned}$$

$$\begin{aligned}
 F4 &= \frac{\text{Accuracy} + \text{Error Tolerancy} + \text{Simplicity}}{3} \\
 &= \frac{(8,59 + 1,48 + 2,1)}{3} \\
 &= \frac{12,17}{3} \\
 &= 4,05
 \end{aligned}$$

Dari hasil yang diperoleh dari perhitungan kemudian nilai kualitas faktor diubah dalam bentuk persentase menggunakan persamaan :

$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Nilai yang didapat}}{\text{Nilai maksimum}} \times 100\% \\
 &= \frac{4,05}{5} \times 100\% \\
 &= 81\%
 \end{aligned}$$

e. *Efficiency*

Execution Efficiency Fa5

$$\begin{aligned}
 &= (w_1c_1 + w_2c_2 + w_3c_3) \\
 &= (0,3 \times 3,5) + (0,3 \times 2,3) + (0,3 \times 2,4) \\
 &= 1,05 + 0,69 + 0,72) \\
 &= 2,46
 \end{aligned}$$

Dari hasil yang diperoleh dari perhitungan kemudian nilai kualitas faktor diubah dalam bentuk persentase menggunakan persamaan :

$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Nilai yang didapat}}{\text{Nilai maksimum}} \times 100\% \\
 &= \frac{2,46}{5} \times 100\% \\
 &= 49,2\%
 \end{aligned}$$

Aspek *functionality* hasil checklist yang diperoleh dari beberapa responden tersebut dilakukan perhitungan persentase dengan persamaan rumus sebagai berikut:

$$\text{Persentase functionality} = \frac{\text{Hasil yang diperoleh}}{\text{Nilai maksimial}} \times 100\%$$

Sehingga total kualitas (Σ) yang diperoleh adalah sebagai berikut :

$$\Sigma = \frac{(0,4 \times Fa1) + (0,3 \times Fa2) + (0,4 \times Fa3) + (0,4 \times Fa4) + (0,3 \times Fa5)}{\text{Nilai Maksimum}} \times 100\%$$

$$= \frac{(0,4 \times 3,3) + (0,3 \times 2,9) + (0,3 \times 2,11) + (0,2 \times 4,05) + (0,2 \times 2,46)}{5} \times 100\%$$

$$= \frac{3,42}{5} \times 100\%$$

$$= 68,4\%$$

Hasil persentase diatas kemudian dibandingkan dengan skala likert, yaitu skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang suatu kejadian . [11]



UNIVERSITAS MIKROSKIL