BAB II
TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem adalah sekelompok unsur yang erat berhubungan satu dengan yang lain yang berfungsi bersama-sama untuk mencapai tujuan tertentu. [3]

Sementara dalam kamus *Webster's Unbridges,* sistem adalah elemen-elemen yang saling berhubungan dan membentuk satu kesatuan atau organisasi. Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain. [4]

Informasi adalah pengetahuan yang diperoleh dari data atau informasi merupakan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. [5]

Atau informasi bisa diartikan juga kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih bermanfaat bagi yang menerima. Jadi ada suatu proses transformasi dari data menjadi suatu informasi, yaitu input proses output. Informasi yang baik adalah informasi yang berkualitas, yaitu informasi harus akurat (bebas dari kesalahan dan tidak menyesatkan), tepat pada waktunya *(up to date),* dan relevan (mempunyai manfaat bagi penerima informasi). [4]

Sistem informasi adalah suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, dimana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan perosedur-prosedur yang terorganisasi atau sistem informasi biasa disebut juga dengan kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen.

Di dalam sistem informasi, manusia berinteraksi dengan manusia, manusia berinteraksi dengan komputer, dan komputer berinteraksi dengan komputer lain. Di dalam sistem informasi, data informasi dan pengetahuan mengalir dibawa oleh dokumen atau media komunikasi elektronik, seperti telepon atau jaringan komputer. Keberadaan sistem informasi diperlukan organisasi untuk mendampingi proses-

proses bisnis dari organisasi. Contohnya proses penjualan *online* yang di dampingi sistem informasi penjualan, yang mencatat pengumpulan data dan informasi tentang penjualan. [3]

Tujuan dari pada sistem informasi yaitu untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan, dan menyajikan sinergi organisasi pada proses. [4]

Manfaat adanya sistem informasi dalam suatu instansi yaitu:

a) Menyajikan informasi guna mendukung pengambilan suatu keputusan.
b) Menyajikan informasi guna mendukung operasi harian.
c) Menyajikan informasi yang berhubungan dengan kepengurusan.

Beberapa komponen sistem informasi dapat diklasifikasikan sebagai berikut [2]:

a) Perangkat keras *(hardware)* dan perangkat lunak *(Software)* yang berfungsi sebagai mesin.
b) Manusia *(people)* dan prosedur *(procedures)* yang merupakan manusia dan tata cara menggunakan mesin.
c) Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.

2.2 Marketplace

Pengertian dari *Online* market adalah segala usaha yang dilakukan untuk melakukan pemasaran suatu produk atau jasa melalui atau menggunakan media internet atau jaringan www, sedangkan *place* sendiri dalam kamus bahasa inggris artinya adalah tempat. Disini dapat disimpulkan pengertian dari *online marketplace* adalah tempat atau wadah untuk melakukan pemasaran produk atau jasa melalui atau menggunakan media internet. [6]

2.3 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah sebuah disiplin yang mengadopsi pendekatan rekayasa seperti metodologi, proses, alat, standar, metode manajemen, sistem jaminan kualitas, dan mengembangkan perangkat lunak skala besar dengan

produktivitas yang tinggi, biaya rendah, kualitas terkontrol, dan pengukuran jadwal pengembangan. [7]

Perangkat lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi preangkat lunak seperti dokumentasi kebutuhan, model sistem, dan cara penggunaannya. Rekayasa perangkat lunak (RPL) merupakan pembangunan sebuah perangkat lunak dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomis yang di percaya dan bekerja secara efisien menggunakan mesin. [7]

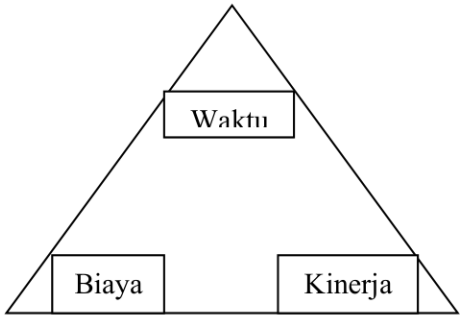
IEEE *Computer society* mendefinisikan rekayasa perangkat lunak sebagai penerapan suatu pendekatan yang sistematis, disiplin atas pengembangan, penggunaan dan pemeliharaan perangkat lunak, serta studi atas pendekatan-pendekatan ini, yaitu penerapan pendekatan engineering atas perangkat lunak. [7]

RPL lebih fokus pada praktek pengembangan perangkat lunak yang bermanfaat bagi
pelanggan (*User*) dengan memenuhi kriteria sebagai berikut [7] :

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan(*Maintainability*).
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*Dependability dan Robust*).
3. Efisiensi dari segi sumber daya dan penggunaan.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan (*Usability*)

Jadi perangkat lunak yang baik adalah perangkat lunak yang fokus kepada pengguna atau pelanggan.

Tujuan rekayasa perangkat lunak secara umum tidak berbeda dengan bidang rekayasa yang lain hal ini dapat kita lihat pada Gambar dibawah ini.



Gambar 2. 1 Tujuan RPL

Gambar 2.1 Tujuan RPL

Dari Gambar di atas dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan *output* yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah [8]:

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. Menghasilkan pereangkat lunak yang kinerjanya tinggi, handal dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis *platform*.
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah.

2.4 Pengujian Perangkat Lunak

Software testing adalah aktivitas-aktivitas yang bertujuan untuk mengevaluasi atribut-atribut atau kemampuan sebuah program atau sistem dan penentuan apakah sesuai dengan hasil yang diharapkan. *Testing* adalah proses pemeriksaan program dengan tujuan tertentu dalam menemukan kesalahan sebelum diserahkan ke pengguna. [9]

Pengujian perangkat lunak merupakan suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji *(under test).* Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya. Teknik-teknik pengujian mencakup, namun tidak terbatas pada, proses mengeksekusi suatu bagian program atau keseluruhan aplikasi dengan tujuan untuk menemukan *bug* perangkat lunak (kesalahan atau cacat lainnya). [9]

Pengujian perangkat lunak dapat dinyatakan sebagai proses *validasi* dan *verifikasi* bahwa sebuah program / aplikasi / produk [9]:

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. [9]

2.4.1 Tahapan *Testing*

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:

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. [9]

2.4.2 Kualitas Sistem Informasi Aplikasi Mobile

Pertanyaan pertama yang muncul ketika membahas pengukuran kualitas perangkat lunak, adalah apa yang sebenarnya yang mau kita ukur. Kualitas perangkat lunak dapat dilihat dari sudut pandang proses pengembangan perangkat lunak (*process)* dan hasil produk yang dihasilkan *(product).* Dan penilaian itu tentu

berorientasi akhir ke bagaimana suatu perangkat lunak dapat dikembangkan sesuai dengan yang diharapkan oleh pengguna. Hal ini berangkat dari pengertian kualitas *(quality)* menurut IEEE standart *Glossary of Software Engineering Technology* yang dikatakan sebagai:

The degree to which a system, component, or process meets customer or user needs or expectation.

Dari sudut pandang produk, pengukuran kualitas perangkat lunak dapat menggunakan standart dari ISO 9126 atau *best practice* yang dikembangkan para praktisi dan pengembang perangkat lunak. [10]

2.5 Metode McCall

Metode McCall merupakan salah satu model yang menjelaskan *Software Quality Factor* atau kualitas perangkat lunak. Model ini memiliki 3 perspektif utama yaitu *product operation* (sifat-sifat operasional dari *software*), *product revision* (kemampuan *software* dalam menjalani perubahan), dan *product transition* (daya adaptasi *software* terhadap lingkungan baru). Prosedur pada McCall terdiri dari menentukan kriteria yang digunakan untuk mengukur suatu faktor, menentukan bobot dari setiap kriteria, menentukan skala nilai setiap kriteria, memasukkan nilai pada tiap kriteria, menghitung nilai total dengan rumus yang ditentukan. *Product operation* meliputi beberapa faktor yaitu *correctness, reliability, usability, integrity, efficiency.* Metode ini memuat kriteria atau faktor kualitas perangkat lunak paling lengkap. Karena metode McCall memiliki ketelitian dan rincian yang baik sehingga dapat digunakan untuk menguji dan menjamin kualitas perangkat lunak sistem informasi. [2]

2.5.1 Product Operations

Sifat-sifat operasional atau *software* berkaitan dengan hal-hal yang harus diperhatikan oleh para perangcang dan pengembang yang secara teknis melakukan penciptaan sebuah aplikasi. Hal-hal yang diukur disini adalah yang berhubungan dengan teknis analisa, perancangan, dan kontruksi sebuah *software.* Faktor-faktor McCall yang berkaitan dengan sifat-sifat operasional *software* adalah:

1. *Correctness* adalah sejauh mana suatu *software* memenuhi spesifikasi dari *mission objective* dari *users.*
2. *Reliability* adalah sejauh mana suatu *software* dapat diharapkan untuk melaksanakan fungsinya dengan ketelitian yang diperlukan.
3. *Efficiency* adalah banyaknya sumber daya komputasi dan kode program yang dibutuhkan suatu *software* untuk melakukan fungsinya.
4. *Integrity* adalah sejauh mana akses ke *software* dan data oleh pihak yang tidak berhak dapat dikendalikan.
5. *Usability* adalah usaha yang diperlukan untuk mempelajari, mengoprasikan, menyiapkan input, dan mengartikan output dari *software.*

.

2.5.2 Product Revision

Setelah sebuah *software* berhasil dikembangkan dan diimplementasikan, akan terdapat berbagai hal yang perlu diperbaiki berdasarkan hasil uji coba maupun evaluasi. Sebuah *software* yang dirancang dan dikembangkan dengan baik, akan dengan mudah dapat direvisi jika diperlukan. Faktor-faktor McCall yang berkaitan dengan kemampuan *software* untuk menjalani perubahan adalah:

1. *Maintainability* adalah usaha yang diperlukan untuk menemukan dan memperbaiki kesalahan *(error)* dalam *software.*
2. *Flexibility* adalah usaha yang diperlukan untuk menemukan dan memperbaiki kesalahan *(error)* dalam *software.*
3. *Testability* adalah usaha yang diperlukan untuk menguji suatu *software* untuk memastikan apakah melakukan fungsi yang dikehendaki atau tidak.

2.5.3 Product Transition

Setelah integritas *software* secara teknis telah diukur dengan menggunakan faktor produk *operational* dan secara implementasi telah disesuaikan dengan faktor produk *revision*, faktor terakhir yang harus diperhatikan adalah faktor transisi yaitu bagaimana *software* tersebut dapat dijalankan pada beberapa *platform* atau kerangka sistem yang beragam. Faktor-faktor McCall yang berkaitan dengan tingkat adaptibilitas *software* terhadap lingkungan baru:

1. *Portability* adalah usaha yang diperlukan untuk mentransfer *software* dari suatu *hardware* atau sistem *software* tertentu agar dapat berfungsi pada *hardware* atau sistem *software* lainnya.
2. *Reusability* adalah sejauh mana suatu *software* (atau bagian *software*) dapat dipergunakan ulang pada aplikasi lainnya.
3. *Interoperability* adalah usaha yang diperlukan untuk menghubungkan satu *software* dengan lainnya. [5]

2.5.4 Metode Perhitungan

1. *Correctness*

Matrik Corectness ini terdapat tiga faktor yang meliputi pengujian kebenaran, matrik itu adalah *Completeness, Consistency,* dan *Tracebility.* Nilai *Correcness* di dapat dengan rumus:

Correcness = Completeness + Consistency + Tracebility

2. *Reliability*

Matrik reliabilitas ini memiliki lima faktor pengujian yaitu *Accuracy, Consistency, Error Tolerance, Modularity,* dan *Simplicity.* Nilai persentase reliabilitas di dapat dengan rumus:

Reliability = Accuracy+Consistency+Error Tolerance+Modularity,+Simplicity

3. *Efficiency*

Matrik *Efficiency* ini memiliki tiga faktor di dalamnya, yaitu *Conciseness, Execution Efficiency,* dan *Operability.* Nilai *Efficiency* di dapat dengan rumus:

Efficiency = Conciseness+Execution Efficiency+Operability

.

4. *Integrity*

Matrik *Integrity* ini memiliki tiga faktor di dalamnya, yaitu: *Auditability, Instrumentation,* dan *Security.* Nilai *Integrity* di dapatkan dengan rumus:

Integrity = Auditability+Instrumentation+Security

5. *Usability*

Matrik *Usability* ini hanya memiliki dua faktor di dalamnya, yaitu *Training, Operability,* dan *Communicativeness.* Nilai *Usability* di dapatkan dengan rumus:

Usability = Training+Operability+Communicativeness [11]