BAB II

KAJIAN PUSTAKA, KERANGKA PEMIKIRAN, DAN HIPOTESIS

2.1. Kajian Pustaka

Kajian pustaka adalah proses umum yang dilakukan peneliti untuk menemukan dan meninjau teori. Dengan tujuan untuk membantu peneliti mengembangkan pengertian serta wawasan yang mendalam tentang hal-hal yang telah dikerjakan serta kecenderungan-kecenderungan yang terjadi. Adapun fungsi dari kajian pustaka adalah sebagai landasan teoritis dalam analisis temuan. Selain itu, tinjauan pustaka juga dapat diartikan sebagai suatu kajian terhadap studi terdahulu yang relevan dengan studi yang dilakukan beberapa penelitian yang telah dituangkan kedalam bentuk buku. Dalam penulisan skripsi ini tidak terlepas dari tinjauan kepustakaan yang berdasarkan pada beberapa sumber.

Berdasarkan uraian diatas, peneliti mengkaji pustaka penelitian ini dengan menentukan variabel yang terkait dari judul penelitian yang diambil. Variabel didefinisikan suatu atribut atau sifat atau nilai dari orang, obyek atau kegiatan yang mempunyai variasi tertentu yang diterapkan oleh peneliti untuk dipelajari dan ditarik kesimpulannya

Variabel yang terdapat pada penelitian ini adalah kualitas *software* Simita sebagai *variable independent* (*variable* X) dan kepuasan *user* sebagai *variable Dependent* (*variable* Y)

2.1.1. Teori Kualitas *Software*

"Karena kualitas *software* merupakan salah satu variabel dalam penelitian ini, maka sudah seharusnya ada teori fundamental yang melandasi pemikiran maupun ide terhadap variabel tersebut. Berikut penjabaran teori dari kualitas *software*:"(Putra, 2013:12)

a. Konsep Kualitas

"Membicarakan tentang konsep kualitas dapat berbeda makna bagi setiap orang, karena kualitas memiliki banyak kriteria dan sangat tergantung pada konsteksnya. Banyak pakar yang mencoba untuk mendefinisikan kualitas berdasarkan sudut pandangnya masing-masing. Misalnya Sule dan Saefullah (2008:363) menyatakan bahwa :

"Kualitas adalah kelengkapan fitur dan manfaat dari sebuah produk yang dihasilkan perusahaan, dan kemampuan produk terebut dalam memenuhi keinginan konsumen. Dengan kata lain, kualitas dapat diukur sampai sejauh mana produk yang dihasilkan sesuai dengan keinginan konsumen."

Sementara Hermawan (2011:10) mengungkapkan bahwa:

"Kualitas merupakan totalitas fitur dan karakteristik yang mampu memuaskan kebutuhan, yang dinyatakan maupun tidak dinyatakan, kualitas mencakup pula daya tahan produk, kehandalan, ketepatan, kemudahan operasi dan perbaikan, serta atribut-atribut nilai lainnya. Beberapa atribut itu dapat diukur secara obyektif. Dari

sudut pandangan pemasaran, kualitas harus diukur sehubungan dengan persepsi kualitas para pembeli."

Tidak ketinggalan, Beberapa teori tentang kualitas yang paling populer adalah yang dikembangkan oleh Deming, Crosby dan Juran (Roza *et al.*, 2012:5). Berikut pemaparan para ahli tersebut:

Menurut Deming, kualitas adalah apapun yang menjadi kebutuhan dan keinginan konsumen. Crosby mempersepsikan kualitas sebagai nihil cacat, kesempurnaan dan kesesuaian terhadap persyaratan. Juran mendefinisikan kualitas sebagai kesesuaian terhadap spesifikasi, jika dilihat dari sudut pandang produsen. Sedangkan secara objektif kualitas menurut Juran adalah suatu standar khusus dimana kemampuannya (*availability*), kinerja (*performance*), kehandalannya (*reliability*), kemudahan pemeliharaan (*maintainability*) dan karakteristiknya dapat diukur.

Jadi berdasarkan pendapat dari berbagai ahli di atas, maka dapat disimpulkan bahwa kualitas merupakan totalitas produk dimana kemampuan *(availability)*, kinerja *(performance)*, kehandalan *(reliability),* kemudahan pemeliharaan *(maintainability)* dan karakteristik produk tersebut dapat diukur sehingga nantinya dapat memuaskan kebutuhan da keinginan konsumen."(Putra,2013:12-13)

b. Perspektif Kualitas

"Berdasarkan konsep kualitas di atas, dapat dilihat para ahli teori kualitas memiliki pandangan yang berbeda-beda mengenai kualitas itu

sendiri. Hal ini merupakan hal yang lumrah karena kualitas sendiri memiliki berbagai macam perspekif. Cortada (Roza *et al.*, 2012:5) berpendapat ada lima macam perspektif mengenai kualitas yaitu:

1) Pendekatan trasedental (*Transcendental Approach*), dimana mutu dapat dirasakan tetapi sulit didefinisikan seperti mutu musik, seni rupa, pelayanan dan lain-lain.
2) Pendekatan dasar produk/hasil (*Product Based Approach*), dimana karakteristik atau atribut yang dapat dikuantisasikan dan dapat diukur.
3) Pendekatan dasar pemakai (*User Based Approach*), tergantung dari pemakai atau orang yang memandangnya sebagai produk yang memuaskan atau tidak memuaskan yang merupakan preferensi seseorang (perceived quality).
4) Pendekatan dasar manufaktur (*Manufacturing Based Approach)*, dimana mutu dipandang atas dasar perekayasaan dan pemanufakturan, sama sebagai persyaratan yang harus ditaati.
5) Pendekatan dasar nilai (*Value Based Approach)*, dimana mutu dipandang dari segi nilai dan harganya. Dengan nilai harga berarti kualitas ini relatif sebab belum tentu produk yang harganya tinggi pasti bermutu tinggi dan sebaliknya produk yang harganya rendah bermutu rendah. “ (Putra, 2013:12-13)

c. Pengertian *Software*

Pengertian *software* secara umum dapat diartikan sebagai kumpulan data-data elektronik yang tersimpan dan diatur oleh komputer, data elektronik yang disimpan oleh komputer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah dan tidak berwujud tidak dapat dilihat oleh mata namun dapat digunakan dan dioperasikan dalam sistem komputer, *software* merupakan kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya. *Software* atau perangkat lunak itu sendiri di bagi menjadi 3 Bagian (Jenis) utama:

1) Sistem Operasi (*operating system*)

   Sistem Operasi adalah *software* pada lapisan pertama yang ditaruh pada memori komputer pada saat komputer dinyalakan. Sedangkan *software-software* lainnya dijalankan setelah Sistem Operasi berjalan, dan Sistem Operasi akan melakukan layanan inti umum untuk *software-softw*are itu. Layanan inti umum tersebut seperti akses ke *disk*, manajemen memori, *scheduling task*, dan antar-muka *user*. Sehingga masing-masing *software* tidak perlu lagi melakukan tugas-tugas inti umum tersebut, karena dapat dilayani dan dilakukan oleh Sistem Operasi.

2) Perangkat lunak bahasa pemrograman (*Programming software*)

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer, adalah teknik komando atau instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu set aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

3) Perangkat lunak aplikasi (Application *software*)

Program aplikasi merupakan perangkat lunak yang dijalankan oleh para pengguna untuk mencapat tujuan tertentu. Contoh kita menjelajah internet dengan menggunakan aplikasi *Browser*. Atau mengedit sebuah berkas dengan aplikasi *Editor.*

d. Kualitas Software

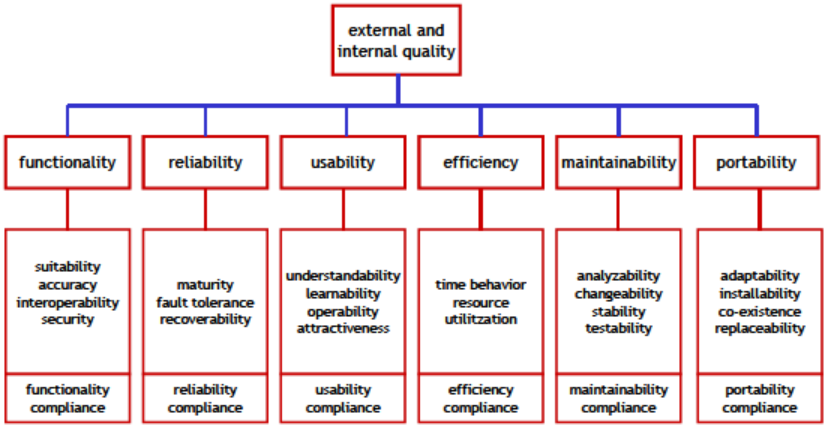
Menurut Roger Presman dalam bukunya Rekayasa Perangkat Lunak (2002: 610) "Kualitas *software* didefinisikan sebagai konfirmasi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang didokumentasikan secara eksplisit dan karakteristik implisit yang

diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional."

Menurut Parwita et al. (2012:91-92), banyaknya penelitian tentang model pengukuran kualitas memang sangat berguna, namun di sisi lain menimbulkan kebingungan karena aspek kualitas yang ditawarkan. Hal ini menyebabkan timbulnya kebutuhan akan suatu model standar. Ini adalah alasan ISO / IEC JTC1 mulai mengembangkan konsensus yang diperlukan dan mendorong standarisasi seluruh dunia. Pertimbangan pertama berasal dari tahun 1978, dan pada tahun 1985 perkembangan ISO / IEC 9126 dimulai. ISO 9126 adalah bagian dari ISO 9000 standar, yang merupakan standar paling penting untuk jaminan kualitas. Dalam model ini, totalitas atribut kualitas produk perangkat lunak diklasifikasikan dalam struktur pohon hirarkis karakteristik dan sub karakteristik. Level tertinggi dari struktur ini terdiri dari karakteristik kualitas dan tingkat terendah terdiri dari kriteria kualitas perangkat lunak.

Model ini menentukan enam karakteristik termasuk *Functionality*, *Reliability, Usability, Efisiensi, Maintainability*, dan *Portabilitas*, yang dibagi lagi menjadi 21 sub karakteristik. Subkarakteristik diwujudkan eksternal ketika perangkat lunak digunakan sebagai bagian dari sistem komputer, dan merupakan hasil dari atribut perangkat lunak internal. Karakteristik didefinisikan berlaku untuk setiap jenis perangkat lunak, termasuk program komputer dan data

yang terdapat dalam *firmware* dan memberikan terminologi yang konsisten untuk kualitas produk perangkat lunak. Model ini juga menyediakan kerangka kerja untuk membuat timbal balik antara kemampuan produk perangkat lunak. Adapun bagan dari Model ISO 9126 ini yaitu :



Gambar 2.1 Bagan Kualitas Software ISO 9126

(Sumber : ISO 9126 (2001, dalam Engels, 2008 : 19))

Berikut ini adalah penjelasan dari setiap karakteistik-karakteristiknya :

1. ***Functionality***

   Kemampuan dari kualitas *software* untuk menyediakan fungsi yang memenuhi kondisi tertentu dan termasuk kebutuhan ketika *software* digunakan dibawah kondisi tertentu.

   a. ***Suitability***

Kemampuan produk *software* untuk menyediakan perangkat sesuai fungsi untuk tugas-tugas tertentu dan tujuan pengguna.

b. *Accuracy*

Kemampuan produk software untuk memberikan hasil yang tepat atau disepakati atau efek dengan tingkat presisi yang diperlukan.

c. *Interoperability*

Kemampuan produk *software* untuk berinteraksi dengan satu atau lebih sistem tertentu.

d. *Security*

Kemampuan produk *software* untuk melindungi informasi dan data, maka dari itu orang yang tidak memiliki autoritas atau sistem tidak dapat membaca atau memodifikasi itu dan orang yang tidak memiliki autoritas atau sistem tidak dijinkan untuk mengakses.

e. *Functionality compliance*

Kemampuan produk *software* untuk mematuhi standar, perjanjian atau peraturan hukum, dan sesuatu yang serupa yang berhubungan dengan fungsi fungsionalitas .

2. *Reliability*

Kemampuan produk *software* untuk mempertahankan tingkat tertentu kinerja bila digunakan dalam kondisi tertentu.

a. *Maturity*

Kemampuan produk *software* untuk menghindari kegagalan sebagai hasil dari kesalahan dalam perangkat lunak.

b. *Fault tolerance*

Kemampuan produk *software* untuk mempertahankan tingkat performansi tertentu dan memulihkan data yang terkena dampak langsung dalam kasus kegagalan .

c. *Recoverability*

Kemampuan produk *software* untuk membangun kembali tingkat kinerja tertentu dan memulihkan data yang terkena dampak langsung dalam kasus kegagalan.

d. *Reliability compliance*

Kemampuan produk *software* untuk mematuhi standar, perjanjian atau peraturan yang berkaitan dengan keandalan.

3. *Usability*

Kemampuan produk *software* untuk dipahami , dipelajari , digunakan dan menarik kepada pengguna, ketika digunakan dalam kondisi tertentu.

a. *Understandability*

Kemampuan produk *software* untuk memungkinkan pengguna untuk memahami apakah *software* cocok, dan bagaimana hal itu dapat digunakan untuk tugas-tugas tertentu dan kondisi penggunaan.

b. *Learnability*

Kemampuan produk *software* untuk memungkinkan pengguna untuk mempelajari penerapannya.

c. *Operability*

Kemampuan produk *software* untuk memungkinkan pengguna untuk mengoperasikan dan mengendalikannya.

d. *Attractiveness*

Kemampuan produk *software* menjadi menarik bagi pengguna.

e. *Usability compliance*

Kemampuan produk *software* untuk mematuhi standar, perjanjian, panduan gaya atau peraturan yang berkaitan dengan penggunaan.

4. *Efficiency*

Kemampuan produk *software* untuk memberikan kinerja yang tepat, relatif dengan jumlah sumber daya yang digunakan, di bawah kondisi yang ditetapkan.

a. *Time behavior*

Kemampuan produk *software* untuk memberikan respon dan pengolahan waktu yang tepat dan tingkat pengambilan saat melakukan fungsinya, di bawah kondisi yang ditetapkan.

b. *Resource utilization*

Kemampuan produk *software* untuk menggunakan jumlah yang tepat dan jenis yang sesuai sumber daya ketika perangkat lunak melakukan fungsinya dalam kondisi yang ditetapkan.

c. *Efficiency compliance*

Kemampuan produk *software* untuk mematuhi standar dan perjanjian yang berkaitan dengan efisiensi.

5. *Maintainability*

Kemampuan produk *software* untuk dimodifikasi. Modifikasi dapat mencakup koreksi, perbaikan atau adaptasi dari perangkat lunak terhadap perubahan lingkungan, dan dalam persyaratan dan spesifikasi fungsional.

a. *Analyzability*

Kemampuan produk *software* untuk dapat didiagnosa kekurangan atau penyebab kegagalan dalam perangkat lunak, atau untuk bagian yang akan dimodifikasi untuk diidentifikasi.

b. *Changeability*

Kemampuan produk *software* untuk memungkinkan modifikasi tertentu untuk diimplementasikan.

c. *Stability*

Kemampuan produk *software* untuk menghindari efek yang tak terduga dari modifikasi *software*.

d. *Testability*

Kemampuan produk *software* untuk memungkinkan modifikasi perangkat lunak untuk divalidasi.

e. ***Maintainability compliance***

Kemampuan produk *software* untuk mematuhi standar atau perjanjian yang berkaitan dengan pemeliharaan.

6. Portability

Kemampuan produk *software* yang akan ditransfer dari satu lingkungan ke lingkungan yang lain.

a. ***Adaptability***

Kemampuan produk *software* yang harus disesuaikan untuk lingkungan tertentu yang berbeda tanpa menerapkan tindakan atau sesuatu yang berarti, selain yang disediakan untuk tujuan ini untuk *software* dipertimbangkan.

b. ***Installability***

Kemampuan produk *software* yang harus diinstal di lingkungan tertentu.

c. ***Co-existence***

Kemampuan produk *software* untuk berdampingan dengan *software* independen lain dalam lingkungan umum membagi sumber daya umum.

d. ***Replaceability***

Kemampuan produk *software* yang akan digunakan di tempat lain yang ditetapkan produk software untuk tujuan yang sama dalam lingkungan yang sama.

e. *Portability compliance*

Kemampuan produk *software* untuk mematuhi standar atau perjanjian yang berkaitan dengan portabilitas.

2.1.2. Teori Kepuasan *User*

"Seperti halnya kualitas *software*, kepuasan *user* juga merupakan salah satu variabel dalam penelitian ini. Oleh karena itu, sudah seharusnya juga terdapat teori fundamental yang melandasi pemikiran maupun ide terhadap variabel ini. Berikut pemaparan dari teori kepuasan *user*: " (Putra, 2013:23)

a. Konsep Kepuasan

"Kepuasan merupakan dampak yang timbul setelah seorang konsumen/*user* menggunakan produk atau sesuatu yang ditawarkan. Biasanya kepuasan tersebut memiliki tingkatan nilai, sesuai dengan pengalaman konsumen/*user* setelah menggunakan suatu produk.

Para ahli juga sudah sejak lama mengamati gejala kepuasan pada konsumen dan dari hasil pengamatan-pengamatan tersebut, muncullah berbagai macam teori tentang kepuasan. Engel *et al.* (Sumarwan *et al.*, 2011:141) mendefinisikan kepuasan sebagai "*As s*

post consumption evaluation that a chosen alternative at least or exceeds expextation".

Sedangkan Buttle *et al.* (Hermawan, 2011:11) mendefinsikan kepuasan sebagai:

"Tanggapan atas pemenuhan pelanggan terhadap sebuah pengalaman konsumsi, atau sebagian kecil dari pengalaman itu. Kepuasan pelanggan merupakan tanggapan yang menyenangkan dari pelanggan karena yang diharapkan terpenuhi, sedangkan ketidakpuasan merupakan tanggapan berupa kekecewaan karena yang diharapkan tidak terpenuhi."

Dari penjabaran mengenai pengertian kepuasan di atas, maka dapat ditarik kesimpulan bahwa kepuasan merupakan tanggapan konsumen/*user* setelah membandingkan kinerja produk dengan harapan yang diinginkan konsumen/*user* tersebut. Tanggapan tersebut biasanya memiliki tingkatan yang beragam berdasarkan pengalaman konsumen/*user* terhadap suatu produk." (Putra, 2013:23-24)

b. Pengukuran Kepuasan Pengguna

Ada 5 dimensi di dalam pengukuran tingkat kepuasan pengguna terhadap penggunaan suatu sistem, di dalam model End-user computing satisfaction model (EUCS) Doll dan Torkzadeh (1988, dalam penelitian Ananda Putra, 2012), yaitu

1. *Content* (isi) dimensi ini mengukur kepuasan pengguna ditinjau dari sisi isi suatu sistem. Isi sistem biasanya berupa fungsi dan model yang dapat digunakan oleh pengguna sistem dan juga informasi yang dihasilkan oleh sistem

2. *Accuracy* (ketepatan) mengukur kepuasan pengguna dari sisi keakuratan data ketika sistem menerima input juga mengolahnya menjadi informasi. Keakuratan sistem diukur dengan melihat seberapa sering sistem menghasilkan output yang salah ketika mengolah input dari pengguna, selain itu dapat dilihat pula seberapa sering terjadi error atau kesalahan dalam proses pengolahan data.

3. *Format* dimensi ini mengukur kepuasan pengguna dari sisi tampilan dan estetika antarmuka sistem, format laporan atau informasi yang dihasilkan oleh sistem apakah antar muka sistem itu menarik dan apakah tampilan sistem memudahkan pengguna ketika menggunakan sistem sehingga secara tidak langsung dapat berpengaruh terhadap tingkat efektivitas dari pengguna

4. *Ease of use* (kemudahan pengguna), mengukur kepuasan pegguna dari sisi kemudahan pengguna atau user friendly dalam menggunakan sistem,
seperti proses memasukan data, mengolah data, dan mencari informasi yang dibutuhkan.

5. *Timeliness* (ketepatan waktu), mengukur kepuasan pengguna dari sisi ketepatan waktu sistem dalam menyajikan atau menyediakan data dan informasi yang dibutuhkan oleh pengguna. Sistem yang tepat waktu dapat dikategorikan real-time, berarti setiap permintaan atau input yang dilakukan oleh pengguna akan langsung di proses dan output akan ditampilkan secara cepat tanpa harus menunggu lama.

2.1.3. Sistem Pendukung Keputusan

Menurut Kusrini (2007:15) bahwa Sistem Pendukung Keputusan/*Decision Support System* (DSS) merupakan sistem informasi yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semistruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter dalam Kusrini, 2007:15-16)

DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi sautu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan. Aplikasi DSS menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat

menggabungkan pemikiran pengambil keputusan. DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. (Kusrini, 2007:16)

Adapun tujuan dari DSS adalah (Turban dalam Kusrini, 2007:16-17) :

1. Membangun manaer atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas kepuasan yang diambil manajer lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi.
5. Peningkatan produktivitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

Menurut Kusrini (2007:19-20) bahwa keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari kestrukturannya yang bisa dibagi menjadi :

1. Keputusan Terstruktur (*Structured Decision*)

   Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas. Keputusan tersebut terutama dilakukan pada manajemen tingkat bawah. Misalnya, keputusan pemesanan barang dan keputusan penagihan piutang.

2. Keputusan Semiterstruktur (*Semistructued Decision*)

   Keputusan semiterstruktur adalah keputusan yang memilki dua sifat. Sebagian keputusan bisa ditangani oleh komputer dan yang lain tetap harus dilakukan oleh pengambil keputusan. Prosedur dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi ada beberapa hal yang masih memerlukan kebijakan dari pengambil keputusan. Biasanya, keputusan semacam ini diambil oleh manajer level menengah dalam suatu organisasi. Contoh keputusan jenis ini adalah pengevaluasian kredit, penjadwalan produksi, dan pengendalian sediaan.

3. Keputusan Tak Terstruktur (*Unstructured Decision*)

   Keputusan tak terstruktur adalah keputusan yang penangannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan tersebut menuntut pengalaman dan

berbagai sumber yang bersifat eksternal. Keputusan tersebut umumnya tersebut pada manajemen tingkat atas. Contohnya adalah keputusan untuk pengembangan teknologi baru, keputusan untuk bergabung dengan perusahaan lain, dan perekrutan eksekutif.

Saat melakukan pemodelan dalam pembangunan DSS dilakukan langkah-langkah sebagai berikut : (Kusrini, 2007:30-31)

1. Studi Kelayakan (*Intelligence*)

   Pada langkah ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah.

   Kepemilikan masalah berkaitan dengan bagian apa yang akan dibangun oleh DSS dan apa tugas dari bagian tersebut sehingga model tersebut bisa relevan dengan kebutuhan si pemilik masalah.

2. Perancangan (*Design*)

   Pada tahapan ini akan diformulasikan model yang akan digunakan dan kriteria-kriteria yang ditentukan. Setelah itu, dicari alternatif model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian, ditentukan variabel-variabel model

3. Pemilihan (*Choice*)

Setelah pada tahap *design* ditentukan berbagai alternatif model beserta variabel-variabelnya, pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya, dilakukan analisis sensitivitas, yakni dengan mengganti beberapa variabel.

4. Membuat DSS

   Setelah menentukan modelnya, berikutnya adalah mengimplementasikannya dalam aplikasi DSS.

a. Model *Analytical Hierarchy Process* (AHP)

Pada dasarnya proses pengambilan keputusan adalah memilih suatu alternatif. Peralatan utama AHP adalah sebuah hierarki fungsional dengan input utamanya adalah persepsi manusia. Keberadaan hierarki memungkinkan dipecahnya masalah kompleks atau tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi bentuk hierarki.

AHP memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga mudah dipahami oleh semua pihakyang terlibat dalam pengambilan keputusan. (Kusrini, 2007 : 133)

b. Prinsip Dasar AHP

Menurut Kusrini (2007 : 133-134), dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah :

1) Membuat Hierarki

   Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen-elemen pendukung, menyusun elemen secara hierarki , dan menggabungkannya atau mensintesisnya.

2) Penilaian kriteria dan alternatif

3) Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty (Kusrini, 2007 : 133), untuk berbagai persoalan, skala 1 sampai 9 adalah skala tebaik untuk mengekspesikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur menggunakan tabel analisis seperti ditunjukan pada tabel berikut :

Tabel 2.1 Skala Penilaian Perbandingan Berpasangan

(Sumber : Kusrini (2007 : 134))

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya.
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya.
5	Elemen yang satu lebih penting daripada

	elemen yang lainnya.
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya.
9	Satu elemen mutlak penting daripada elemen lainnya.
2, 4, 6, 8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan.
Kebalikan	Jika aktivitas i mendapat satu angka dibandingkan dengan aktivitas j, maka j memiliki nilai kebalikannya dibandingkan dengan i.

4) *Synthesis of priority* (menentukan prioritas)

   Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*pairwase comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgement* yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.

5) *Logical consistency* (Konsistensi Logika)

   Konsistensi memiliki 2 makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan

relevansi. Kedua, menyangkut tingkat hubungan antarobjek yang didasarkan pada kriteria tertentu.

c. Prosedur AHP

Menurut Kusrini (2007 : 135-137), prosedur atau langkah-langkah dalam metode AHP meliputi :

1) Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakansasaran sistem secara keseluruhanpada level teratas.
2) Menentukan prioritas elemen.
   a. Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.
   b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk mempresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya.
3) Sintesis

   Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah :

   a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.

b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4) Mengukur konsistensi

Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :

a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
b. Jumlahkan setiap baris.
c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
d. Jumlahkan hasil bagi diatas dengan banyaknyaelemen yang ada, hasilnya disebut λ maks.

5) Hitung *Consistency Index* (CI) dengan rumus :

$$CI = (\lambda \text{ maks-n})/n$$

Dimana n banyaknya elemen

6) Hitung rasio Konsistensi / *Consistency Ratio* (CR) dengan rumus :

$$CR = CI/IR$$

Dimana CR = *Consistency Ratio*, CI = *Consistency Index*, IR = *Index Random Consistency*

7) Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data *judgment* harus diperbaiki. Namun, jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar.

Daftar Indeks Random Konsistensi bisa dilihat dalam tabel berikut ini :

Tabel 2.2 Daftar Indeks random Konsistensi

(Sumber : Kusrini (2007:136-137))

Ukuran Matriks	Nilai IR
1,2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45

10	1,49
11	1,51
12	1,48
13	1,56
14	1,57
15	1,59

2.1.4. *Maturity Model*

Maturity model digunakan sebagai *metric* untuk mengukur tingkat pengembangan sistem informasi. *Maturity model* untuk setiap proses COBIT, digunakan untuk mengindentifikasi:

1. Kinerja aktual perusahaan – Dimana perusahaan ini
2. Status industri yang terkini – Sebagai perbandingan
3. Target peningkatan perusahaan – Posisi yang ingin dicapai perusahaan

Menurut ITGI (2010), kapabilitas proses manajemen tidak sama untuk setiap proses kinerja. Kapabilitas yang diperlukan ditentukan oleh tujuan bisnis dan tujuan TI. Untuk mengukur kapabilitas kinerja TI, COBIT menggunakan lima *level maturity model*:

1. Skala 0 : *Non-Existent.*

   Tidak ada proses yang dapat dikenali. Manajemen bahkan tidak menyadari adanya isu pengelolaan yang ditangani

2. Skala 1 : *Initial.*

Terdapat bukti bahwa perusahaan telah mengetahui adanya isu-isu TI yang harus ditangani. Tidak ada proses yang standar dan pada umumnya menggunakan pendekatan *ad hoc* atau *case by case basis.* Secara keseluruhan pendekatan yang digunakan dalam pengelolaan tidak terorganisir

3. Skala 2 : *Repeatable.*

   Proses dilengkapi dengan prosedur yang diikuti oleh individu-individu yang memiliki kesamaan tugas. Tidak ada program pelatihan secara formal, yang bertujuan untuk mengkomunikasikan prosedur-prosedur dan tanggung jawab setiap individu. Proses sangat bergantung pada keahlian setiap individu.

4. Sekala 3 : *Defined*.

   Proses dilengkapi dengan prosedur-prosedur yang terstandarisasi, terdokumentasi dan dikomunikasikan melalui pelatihan secara formal. Walaupun demikian penyimpangan terhadap ketaatan pada prosedur masih sulit untuk dideteksi. Prosedur-prosedur yang dibuat merupakan fomalisasi dari kegiatan-kegiatan yang ada.

5. Skala 4 : *Managed and measurable*.

   Proses pengawasan dan penilaian ketaatan pada prosedur sudah ditetapkan dan terdapat aktifitas untuk melakukan proses perbaikan ketika proses berjalan tidak efektif.

6. Skala 5 : *Optimised*

   Proses yang ada telah disempurnakan praktek yang lebih baik,

berdasarkan hasil perbaikan terus menerus. TI digunakan secara terinegrasi untuk mengoptimalkan proses kerja, mnyediakan alat untuk meningkatkan kualitas dan efektivitas serta membuat perusahaan mudah beradaptasi.



Gambar 2.2 Gambar maturity model COBIT

(Sumber: ITGI, 2010)

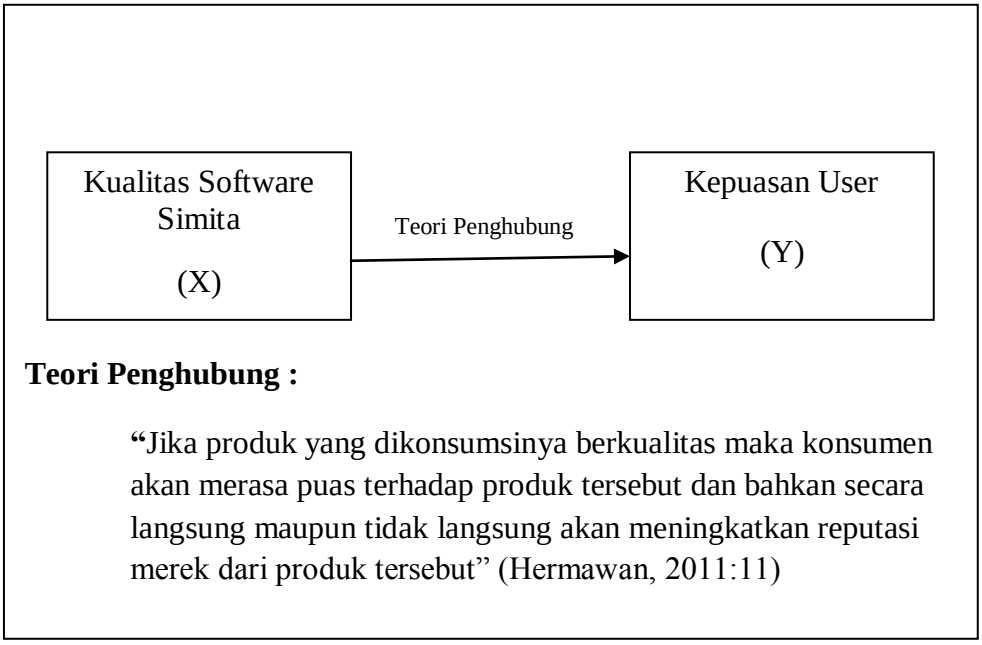
Terdapat 5 (lima) jenis kemungkinan responden, dikaitkan dengan maturity model yang direkomendasikan oleh COBIT (skala 0-5). Responden akan memilih tingkat aktivitas yang sesuai dengan kondisi perusahaan saat ini. Maturity model akan membantu para professional menjelaskan kepada manajer tentang kekurangan manajemen TI dan menetap target yang mereka perlukan. Tingkat mataruty akan dipengaruhi oleh sasaran bisnis perusahaan dan oprasi lingkungan

2.1.5. Keterkaitan Antar Variabel

"Pada penelitian ini terdapat satu variabel bebas (X) dan satu variabel terikat (Y). Variabel bebasnya adalah kualitas *software* Simita dan variabel terikatnya berupa kepuasan *user*. Dalam penelitian ini, teori penghubung yang digunakan antara variabel kualitas *software*

Simita (X) dengan variabel kepuasan *user* (Y) yaitu pada jurnal yang ditulis oleh Hermawan (2011:11), yang berbunyi "Jika produk yang dikonsumsinya berkualitas maka konsumen akan merasa puas terhadap produk tersebut dan bahkan secara langsung maupun tidak langsung akan meningkatkan reputasi merek dari produk tersebut".

Agar lebih jelas, berikut diberikan gambaran teori penghubung antara kualitas *software* Simita dengan kepuasan *user."* (Putra, 2013:37)



Gambar 2.3 Keterkaitan Antara Variabel *Independent* dan Variabel *Dependent*

(Sumber : Putra, 2013:30, dimodifikasi)

2.2. Kerangka Pemikiran

"Menurut Sekaran (Sugiyono, 2010:91), kerangka berfikir merupakan "model konseptual tentang bagaimana teori berhubungan

dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting”

Kerangka pemikiran yang baik akan menjelaskan secara teoritis pertautan antara variabel yang akan diteliti. Jadi secara teoritis perlu dijelaskan antar variabel independen dan dependen. Pertautan antar variabel tersebut selanjutnya dirumuskan ke dalam bentuk paradigma penelitian (berupa diagram) yang didasarkan pada kerangka pemikiran.” (Putra, 2013:38)

a. Naratif

“Kualitas dapat disimpulkan sebagai totalitas produk dimana kemampuan *(availability)*, kinerja *(performance)*, kehandalan *(reliability)*, kemudahan pemeliharaan *(maintainability)* dan karakteristik produk tersebut dapat diukur sehingga nantinya dapat memuaskan kebutuhan dan keinginan konsumen. Sedangkan kepuasan merupakan tanggapan konsumen/*user* setelah membandingkan kinerja produk dengan harapan yang diinginkan konsumen/user tersebut. Tanggapan tersebut biasanya memiliki tingkatan yang beragam berdasarkan pengalaman konsumen/*user* terhadap suatu produk.” (Putra, 2013:39)

Untuk melihat kualitas *software* Simita pada Program Studi Sistem Informasi Unikom, maka mau tidak mau harus dilakukan pengukuran terhadap kualitas *website* tersebut agar didapatkan hasil secara ilmiah. Dalam mengukur kualitas *software* tentunya harus ada karakteristik-

karakteristik yang menjadi patokan dalam pengukuran kualitas tersebut. Di sini peneliti mengambil karakteristik tersebut dari kriteria *software* ISO 9126, yaitu: *funcionality, reliability, usability, efficiency, maintanability,* dan *portability* (Engels 2008:18).

Seperti halnya kualitas, dalam penelitian yang menyangkut kepuasan *user* pun, sudah seharusnya juga ada karakteristik untuk melihat apa saja yang menjadi patokan dalam meneliti kepuasan. Indikator-indikator kepuasan *user* peneliti dasarkan pada *key* atau kunci penting dari teori tentang model End-user computing satisfaction (EUCS) Doll dan Torkzadeh. Pada Model tersebut dijelaskan bahwa terdapat 5 dimensi di dalam pengukuran tingkat kepuasan pengguna terhadap penggunaan suatu sistem,yaitu *content, ease of Use, Format, Accuracy, dan Timeliness*.

b. Perbedaan dengan Penelitian Sebelumnya

Perbedaan penelitian ini dengan penelitian sebelumnya, dijelaskan pada tabel dibawah ini :

Tabel 2.3 Perbedaan dengan Peneliti Terdahulu

No	Nama Peneliti dan Tahun Penelitian	Judul	Hasil Penelitian	Perbedaan
1	Putra, A. S. (2013)	Kontribusi Kualitas Website	1. Kualitas *Website* SMA Negeri 6 Bandung yang	1. Penggunaan Teori kualitas

		SMA negeri 6 andung Terhadap Kepuasan User dengan Menggunakan Metode Analytical Hierarchy Process (AHP)	sedang diimplementasikan, dapat dikatakan berkontribusi terhadap kepuasan *user*. Hal ini dapat dilihat dari jawaban yang diberikan responden dalam kuesioner penelitian dan hasil olah data. Akan tetapi implementasi tersebut belum dapat membuat *user* puas. Hal ini dapat disebabkan oleh berbagai fenomena atau	software

			permasalahan yang nyata keberadaannya dan banyak dialami oleh *user*. 2. Tanggapan responden terhadap kualitas *Website* SMA Negeri 6 Bandung dikategorikan cukup baik. 3. *User* merasa belum puas terhadap implementasi kualitas *website* SMA Negeri 6	

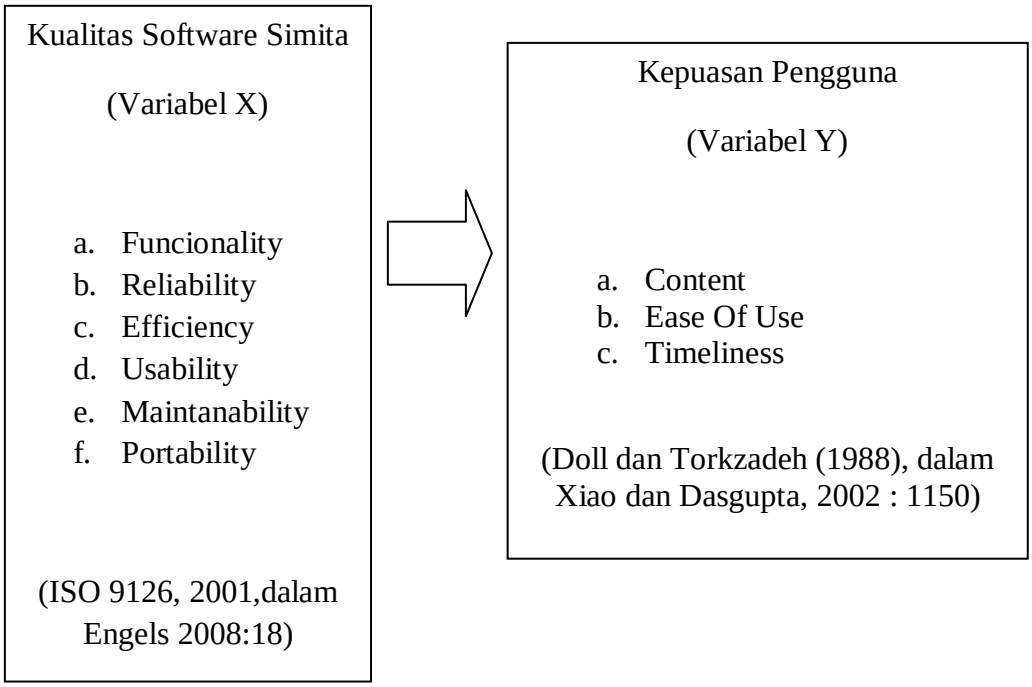
			Bandung. Hal ini terlihat dari rata-rata tingkat kepuasan *user* terhadap implementasi kualitas *website* secara keseluruhan, yang nilainya sebesar 0,727. 4. Dari hasil perhitungan dan olah data, didapat hasil korelasi pearson sebesar 0,571. Ini menandakan bahwa korelasi antara variabel *independent* (kualitas *website*	

			SMA Negeri 6 Bandung) dengan variabel *dependent* (kepuasan *user*) cukup kuat. Pada penelitian ini diperoleh koefisien determinasi sebesar 32,6%. Ini mengindikasikan bahwa variabel *independent* tersebut berkontribusi sebesar 32,6% terhadap variabel *dependent.*	
2	Lee, John et. Al (2008)	*Using Analytical Hierarchy*	1. Faktor yang terpenting dalam *Smartphone*	1. Segi eksplanasi penelitian

		Process to Evaluate Software Quality Characteristics of Smartphone	adalah *reliability*. 2. Untuk semua kriteria, kualitas dari suara juga merupakan faktor yang signifikan dalam kualitas *software*. 3. optimasi infrastruktur software dan pengurangan efek samping dari modifikasi program perlu menjadi perhatian juga.	2. Variabel yang digunakan hanya 1 variabel yaitu hanya kualitas *software* saja

c. Bagan Kerangka Pemikiran

Berdasarkan uraian di atas maka dirumuskan paradigma mengenai kontribusi kualitas *software* Simita Program Studi Sistem Informasi Unikom terhadap kepuasan *user* dalam bentuk bagan kerangka pemikiran sebagai berikut :



Gambar 2.4 Bagan Kerangka Pemikiran Pengaruh Kualitas *Software* Simita terhadap kepuasan *user*

Pada penelitian ini, seperti yang telah dipaparkan pada bagian batasan masalah di BAB I, dari enam karakteristik kualitas *software* pada bagan kerangka pemikiran, hanya dua karakteristik yang diteliti. Ada pun dua karakteristik tersebut yaitu *maintanability* dan *funcionality*. Hasil perhitungan AHP dan pengambilan keputusan dapat dilihat pada bagian lampiran.

2.3. Hipotesis

"Seorang peneliti perlu merumuskan jawaban atau dugaan sementara yang kebenarannya kelak akan diuji serta menjadi acuan dalam

proses penelitian yang dilakukan. Jawaban atau dugaan sementara ini disebut dengan hipotesis. Sebagaimana yang diungkapkan Arikunto (2002:64) bahwa hipotesis merupakan "Suatu jawaban yang bersifat sementara terhadap permasalahan penelitian, sampai terbukti melalui data yang terkumpul". Ada pun hipotesis dalam penelitian ini dirumuskan sebagai berikut: " (Putra, 2013:44-45)

a. Hipotesis nol (H0)

   Tidak terdapat hubungan yang signifikan antara Kualitas *Software* Simita Program Studi Sistem Informasi Unikom (X) dengan kepuasan *user* (Y).

b. Hipotesis penelitian (H1)

   Terdapat hubungan yang signifikan antara Kualitas *Software* Simita Program Studi Sistem Informasi Unikom (X) dengan kepuasan *user* (Y).