

PENGGUNAAN STANDARD ISO 9126 UNTUK MENGEVALUASI KEEFEKTIFAN PERANGKAT LUNAK

Agus Sukoco¹⁾

¹⁾Program Studi Teknik Informatika

Fakultas Ilmu Komputer

Universitas Bandar Lampung

Jln. Z.A. Pagar Alam No.26 Labuhan Ratu Bandar Lampung 35142

Telp. (0721) 701463, (0721) 701979 Fax. (0721) 701467 Web. www.ubl.ac.id

E-mail: kcoo16@yahoo.com¹⁾

Handphone: 081514356424¹⁾

Abstrak

Mengevaluasi perangkat lunak yang telah diimplementasikan dalam sebuah perusahaan merupakan elemen penting dari sebuah penjaminan proses pengembangan rekayasa perangkat lunak dewasa ini. Dengan demikian perusahaan yang telah berintegrasi dengan perangkat lunak di dalam proses bisnisnya, harus mengevaluasi perangkat lunak dengan memantau serta mengukur karakteristik perangkat lunak yang dihasilkan untuk membuktikan bahwa persyaratan perangkat lunak yang diinginkan telah dipenuhi penggunaannya. Maka, berkaitan dengan hal tersebut diperlukan pengukuran dan pemantauan berupa evaluasi perangkat lunak yang efektif yang dihasilkan sesuai dengan standard yang telah ada.

Salah satu standard yang bisa digunakan untuk mengevaluasi kualitas software yaitu Standard ISO 9126. Standard ini merupakan salah satu frame work umum mengenai karakteristik dari kualitas perangkat lunak, yang dipercaya mempunyai kekuatan yang lebih adaptable yang dapat digunakan untuk seluruh sistem, terutama untuk menetapkan kerangka umum dalam mengevaluasi sebuah software. Sehingga pihak manajemen akan lebih akurat dalam memperoleh informasi untuk menentukan keputusan strategis.

Kata Kunci : ISO 9126, Aplikasi Wizard, Pengukuran Software.

1. PENDAHULUAN

Pemanfaatan teknologi informasi sangat berkaitan erat dalam pengembangan bisnis dewasa ini, faktor penting dalam perkembangan tersebut juga sangat dipengaruhi oleh kualitas perangkat lunak (*Software*), terutama sebagai interaksi pengguna dengan perangkat keras (*hardware*). Sehingga dalam bisnis prosesnya, ada beberapa perusahaan yang mulai bergantung dengan perangkat lunak. Saat ini perangkat lunak sudah menjadi kekuatan yang menentukan, dan menjadi mesin yang mengendalikan pengambilan keputusan di dalam dunia bisnis (Pressman, 2003 : p2-3). Pengelolaan perangkat lunak sebagai pengambilan keputusan yang strategis, tentunya harus ditunjang oleh sistem dan perangkat lunak yang berkualitas. Maka

diharapkan *management* dapat memperoleh informasi sebagai sumberdaya yang strategis dan informasi yang berkualitas.

Agar perangkat lunak dapat terjamin dengan baik, maka diperlukan pengendalian dan pengelolaan yang mengacu pada kualitas perangkat lunak. Jaminan kualitas perangkat lunak merupakan aktivitas mendasar bagi banyak bisnis yang menghasilkan produk yang akan digunakan oleh user baik internal perusahaan maupun eksternal. Untuk memperoleh kualitas perangkat lunak yang diharapkan, mengevaluasi kualitas produk suatu perangkat lunak merupakan elemen kritis dari jaminan perangkat lunak sehingga dapat merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean (Pressman, 2003 : p216-217).

Pengendalian untuk memperoleh kualitas perangkat lunak yang sesuai diharapkan dengan visi, misi dan tujuan organisasi merupakan elemen kritis bagi perusahaan. Sehingga pengujian perangkat lunak yang mengacu pada standard tertentu patut dilakukan. Terutama bagi perusahaan yang sudah menerapkan sistem manajemen mutu yang mengacu pada standar ISO 9001:2008, di dalam standard ISO 9001:2008 klausul 6.3 (b) tertulis bahwa suatu perusahaan harus memperhatikan dalam penyediaan dan pemeliharaan infrastruktur *software*. Pemeliharaan *software* yang dikaitkan dengan klausul 8.2.4, yakni organisasi harus memantau dan mengukur karakteristik produk yang dihasilkan untuk membuktikan bahwa persyaratan produk telah dipenuhi. Maka, berkaitan dengan hal tersebut diperlukan pengukuran dan pemantauan berupa evaluasi *software* yang efektif yang dihasilkan sesuai dengan standard yang telah ada.

Salah satu standard yang bisa digunakan untuk mengevaluasi kualitas *software* yaitu Standard ISO 9126. *Standard* ini merupakan salah satu *frame work* umum mengenai karakteristik dari kualitas perangkat lunak, yang dipercaya mempunyai kekuatan yang lebih *adaptable* yang dapat digunakan untuk seluruh sistem, terutama untuk menetapkan kerangka umum dalam mengevaluasi sebuah *software* (Bee Bee Chua, 2004 : p185), sehingga dapat mengevaluasi keefektifan dan kualitas perangkat lunak, dan mendeteksi kesalahan potensial, sehingga visibilitas perangkat lunak suatu elemen sistem dan biaya yang muncul akibat kegagalan perangkat lunak, dapat ditekan.

PT. SMI yang telah mengimplementasikan ISO 9001 : 2008, berdasarkan acuan standardnya terutama untuk peningkatan yang berkelanjutan, khususnya di divisi IT dituntut memperoleh andil dalam peningkatan tersebut. Untuk memenuhi peningkatan yang berkelanjutan tersebut, divisi IT yang menghasilkan perangkat lunak harus memantau dan mengukur produk yang telah dihasilkan. Divisi IT harus memantau dan mengukur karakteristik produk untuk membuktikan bahwa perangkat lunak yang dihasilkan telah memenuhi persyaratan, sehingga perencanaan produk yang sesuai

yang direncanakan akan tercapai. Dengan demikian persyaratan ISO 9001:2008 dan sasaran mutu dari divisi IT dapat terpenuhi dengan baik.

Dari hasil internal audit yang dilakukan secara teratur dan terencana, terdapat temuan yang sampai saat ini belum diselesaikan, sehingga karena temuan tersebut telah ditetapkan sebagai temuan yang harus di agendakan dalam rapat tinjauan manajemen, temuan tersebut yaitu pengukuran dan pemantauan dalam hal ini mengevaluasi keefektifan *software* yang telah diimplementasikan oleh divisi IT. Sehingga keefektifan dan kualitas dari perangkat lunak pada saat ini belum dapat diketahui kehandalannya. Hal ini menyulitkan untuk mengevaluasi perangkat lunak secara produk secara periodik dengan sistem yang efektif dan efisien.

Oleh karena itu, diperlukan alat bantu untuk mempermudah melakukan pengujian *software* tersebut, sehingga diperlukan pengembangan sebuah aplikasi yang mampu mendukung aktivitas evaluasi dari pengukuran kualitas perangkat lunak berdasarkan ISO 9126. Sehingga pihak manajemen akan lebih akurat dalam memperoleh informasi untuk menentukan keputusan strategis.

Aplikasi *wizard* merupakan *tools* yang khas atau unik digunakan dalam proses pengembangan struktur yang baik, dengan menampilkan beberapa pertanyaan dan jawaban. Dan setiap langkah dalam *wizard* dimungkinkan dengan tampilan *user interface* yang baik (Burton, 2000 : p2), sehingga mempermudah dalam pelaksanaan evaluasi *software* sistem informasi.

1.1 Identifikasi Masalah

Berdasarkan kenyataan yang telah di sebutkan sebelumnya, maka identifikasi masalah dari penelitian ini dapat di rumuskan sebagai berikut:

- a. Apa standard acuan yang diperlukan untuk evaluasi perangkat lunak?
- b. Apakah Perangkat lunak yang digunakan sudah pernah dilakukan evaluasi dengan standard tertentu?

1.2 Ruang Lingkup Masalah

Mengingat akan keterbatasan waktu, dan biaya maka ruang lingkup penelitian ini dibatasi pada Standard ISO 9126 merupakan acuan yang dapat digunakan untuk mengukur kualitas perangkat lunak di PT. SMI, khususnya untuk Program Penjualan.

1.3 Perumusan Masalah

Perumusan masalah ini adalah Bagaimana menyusun aplikasi untuk mengevaluasi perangkat lunak di PT. SMI dengan menggunakan Standard ISO 9126 ?

1.4 Tujuan dan Manfaat Penelitian

Tujuan Penelitian ini adalah:

- a. Membangun aplikasi untuk evaluasi perangkat lunak dengan menggunakan Standard ISO 9126.
- b. Mempermudah melakukan pengukuran terhadap kualitas perangkat lunak.

Sedangkan manfaat dari penelitian ini dari beberapa segi yakni:

- a. Memberikan manfaat praktis adalah hasil dari penelitian ini diharapkan bermanfaat bagi tim yang mempunyai tujuan untuk mengevaluasi *software* yang telah diimplementasikan di perusahaannya.
- b. Memberikan manfaat teoritis semoga dapat memberikan sumbangan bagi pengembangan teori pendidikan. Terutama dalam pengembangan *quality assurance* khususnya dibidang rekayasa perangkat lunak.
- c. Diperolehnya informasi dengan evaluasi tinggi dengan kualitas perangkat lunak yang baik.

2. LANDASAN TEORI

2.1 Jaminan Kualitas Perangkat Lunak

Jaminan kualitas perangkat lunak merupakan suatu aktivitas perlindungan pada suatu proses secara keseluruhan dalam pengembangan perangkat lunak.

2.2 Kualitas Perangkat Lunak

American Heritage Dictionary mendefinisikan kata kualitas sebagai "sebuah karakteristik atau atribut dari sesuatu." sebagai atribut, kualitas mengacu pada karakteristik yang dapat diukur, sesuatu yang dapat dibandingkan dengan standar yang sudah diketahui.

2.3 Pengukuran Produk Perangkat Lunak

Pengukuran perangkat lunak merupakan tema kajian yang turun temurun dalam sejarah rekayasa perangkat lunak (*Software Engineering*). Kajian tersebut dapat diukur dari segi proses maupun produk.

2.4 ISO 9126

Salah satu standar kualitas untuk mengukur kualitas produk yang dihasilkan adalah ISO 9126. standar ISO 9126 terbagi menjadi 4 (empat) bagian, yakni : model kualitas, internal metric, external metric dan metric kualitas.

Enam karakteristik dari model kualitas *software* adalah:

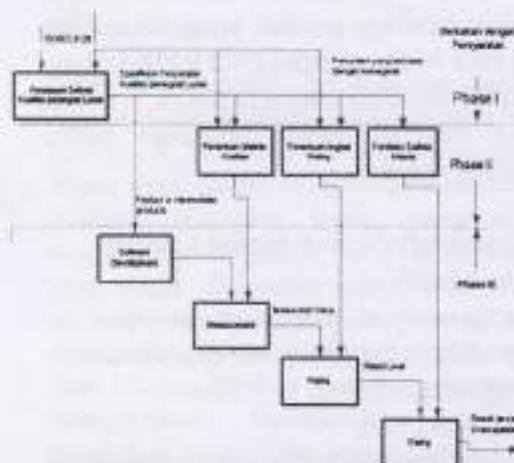
1. *functionality* yaitu kemampuan dari segi fungsi produk perangkat lunak yang menyediakan kepuasan kebutuhan user.
2. *Reliability* yaitu kemampuan perangkat lunak untuk perawatan dengan level performansi.
3. *Usability* yaitu atribut yang menunjukkan tingkat kemudahan pengoperasian perangkat lunak.
4. *Efisiensi* yaitu menyangkut waktu eksekusi dan kemampuan yang berhubungan dengan sumber daya fisik yang digunakan ketika perangkat lunak dijalankan.
5. *Maintability* yaitu tingkat kemudahan perangkat lunak tersebut dalam mengakomodasi perubahan-perubahan.
6. *Portability* yaitu kemampuan yang berhubungan dengan kemampuan perangkat lunak yang dikirim ke lingkungan berbeda.

2.4 Aplikasi Wizard

Design wizard merupakan *tool software* yang menyatukan informasi yang berkaitan dengan *reference architecture*, implementasi dari komponennya dan *interrelations*, batasan-batasan yang dipergunakan dalam tool sesuai dengan komposisi yang sah. *Design* mempergunakan *wizard* merupakan *generator tools* dalam mengotomasi sebuah aplikasi yang berkaitan dengan *system families* (Guillermo, 2000 : p3).

2.5 Kerangka Pemikiran

Kerangka pemikiran model proses pengukuran perangkat lunak menurut standard ISO 9126 dapat diuraikan pada gambar di bawah ini.



Gambar 1. Model Pengukuran Proses Pengukuran Perangkat Lunak

Gambar di atas menjelaskan mengenai langkah-langkah secara dalam pengukuran kualitas dari suatu perangkat lunak, yang terdiri dari 3 (phase) yaitu:

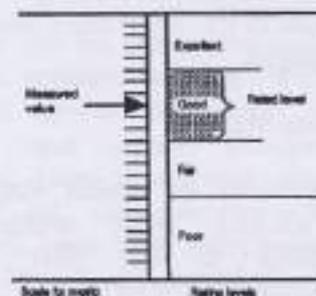
a. Phase I

Menentukan Kriteria/Karakteristik yang digunakan untuk mengukur kualitas perangkat lunak, dalam hal ini mengacu pada standard ISO 9126 dan persyaratan managerial yang telah ditetapkan pada dasar pembuatan *software*.

b. Phase II

Phase II yaitu menentukan bobot/Skala Metric dari masing-masing karakteristik maupun sub karakteristiknya. Menentukan skala dari nilai

kriteria sehingga diperoleh *Rating Level* dari hasil pengukuran. Menentukan Nilai dari tiap-tiap kriteria, dalam hal ini adalah Buruk, Sedang, Bagus dan bagus sekali seperti gambar di bawah ini:

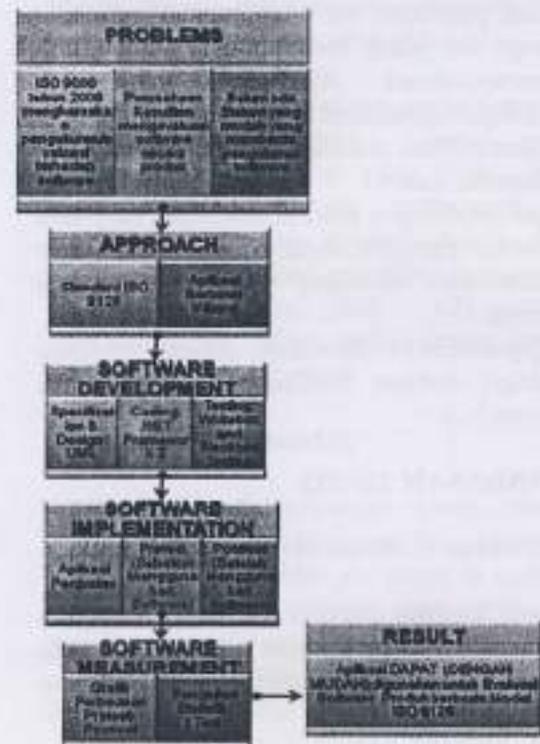


Gambar 2. Penentuan Bobot Penilaian

c. Phase III

Phase III melakukan pengukuran dan evaluasi dari tiap-tiap karakteristik yang telah ditentukan di Phase II. Sehingga dapat ditentukan hasil pengukuran dan evaluasinya.

Sedangkan kerangka pemikiran dalam proses kegiatan (*framework*) pengukuran kualitas perangkat lunak dapat dilihat gambar di bawah ini:



Gambar 3 Proses Kegiatan (Framework) Pengukuran Kualitas Perangkat Lunak

3. METODE PENELITIAN

3.1 Permodelan menggunakan Use Case Diagram

Permodelan menggunakan *use case* diagram akan menyajikan interaksi antara *use case* dan *actor*. Dimana *actor* dapat berupa user, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun.

Use case diagram menggambarkan persyaratan-persyaratan yang harus dipenuhi sistem dalam hal ini pengembangan aplikasi *wizard* dari pandangan pemakai. Maka langkah-langkah yang dilakukan dalam mengembangkan aplikasi *wizard* sebagai berikut:

3.2 Mengidentifikasi pelaku bisnis

Pelaku dalam hal ini disebut sebagai Aktor yakni seseorang yang berhubungan dengan aplikasi *wizard*. Aktor dalam pengguna aplikasi *wizard* hanya seorang user saja. User dalam pengguna aplikasi *wizard* bisa seorang internal auditor IT, atau seorang yang ditugaskan dalam mengevaluasi sebuah aplikasi yang sudah diimplementasikan. Seorang Internal Auditor mempunyai tugas untuk mengukur dan mengevaluasi aplikasi yang telah dikembangkan. Wewenang dari seorang internal auditor adalah menentukan dan memberikan nilai.



Gambar 4. Use Case Pengukuran Kualitas Software

Gambar di atas menerangkan bahwa seorang user akan mengukur kualitas sebuah *software* dari karakteristik fungsi, reliabilitas, usabilitas, portabilitas, maintainabilitas, efisiensi dan menganalisis hasil pengukuran evaluasi dari sebuah *software*, dalam bentuk aplikasi *wizard*. Dari masing-masing karakteristik, kemudian dijelaskan ke dalam subkarakteristik pada *use case* diagram.

Langkah selanjutnya adalah Internal Auditor melakukan pengukuran *software* dengan menentukan bobot/Skala Metric dari masing-masing karakteristik maupun sub karakteristiknya. Menentukan skala dari nilai kriteria sehingga diperoleh *Rating Level* dari hasil pengukuran. *Use case* tersebut menjelaskan pengukuran *software* dari segi fungsi sebuah *software*. Bagaimana fungsi sebuah *software* yang telah dibuat memberi kepuasan kepada penggunanya.

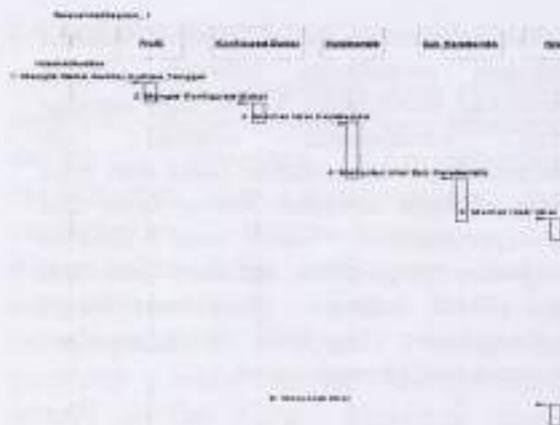
3.3 Permodelan Menggunakan Class Diagram

Setelah dibuat *use case diagram*, langkah selanjutnya adalah membuat *Class Diagram* berdasarkan *use case diagram* tersebut. *Class diagram* merupakan obyek-obyek yang terdapat dalam aplikasi *wizard*. *Class diagram* terdapat *class* konfigurasi bobot, profil, karakteristik dan laporan. *Class diagram* dari aplikasi yang dibuat digambarkan seperti di bawah ini:



Gambar 5. Class Diagram

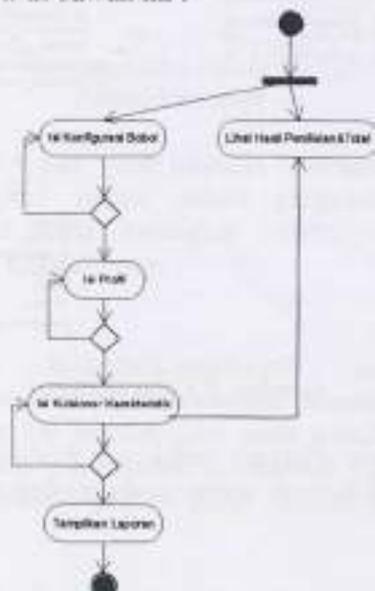
Sequence diagram untuk set modul kualitas perangkat lunak seperti pada gambar di bawah ini:



Gambar 6. Sequence Diagram untuk mengukur kualitas perangkat lunak

Diagram sekuensial yang menggambarkan obyek dengan interaksi obyek tersebut, dalam hal ini seorang user akan membuka wizard suitability yang berisi pertanyaan dan bobot nilai yang akan diberikan oleh user hasil dan mengisi sesuai dari hasil penilaian aplikasi yang diukur atau dievaluasi. Langkah awal adalah menuju ke Profil untuk membuat proyek baru, sampai mengukur nilai dan outputnya adalah laporan.

Activity diagram merupakan teknik mendiskripsikan logika prosedural, aliran kerja dalam mengukur perangkat lunak. Berikut activity diagram untuk mengukur perangkat lunak, seperti di bawah ini :



Gambar 7. Activity Diagram Mengevaluasi Perangkat Lunak

3.4 Teknik Analisis

Proses Perbandingan Hasil Pre-Test dan Post-Test

Pada bagian ini akan dilakukan proses perbandingan hasil pengukuran penelitian yang telah didapatkan sebelumnya. Adapun metode perbandingan ini adalah dengan Uji *T-Test (Paired Sample T Test)*. Metode ini digunakan karena *t-test* dapat digunakan untuk menguji kecocokan atas perbedaan pada suatu eksperimen yang menggunakan satu kelompok sampel. Apabila sebelum melakukan pengukuran eksperimen, peneliti melakukan pengukuran awal (*pre test*), maka peneliti akan mempunyai dua kelompok nilai yang berasal dari satu kelompok sampel. Apabila eksperimen itu mempunyai dampak terhadap hasil (tujuan eksperimen), maka kedua kelompok skor tersebut akan menunjukkan perbedaan yang signifikan. Apabila hasil perhitungan tersebut berbeda secara signifikan, maka hipotesa diterima. Langkah-langkah pengujian sebagai berikut :

a. Menentukan Hipotesis

H_0 : Tidak ada perbedaan antara pre-test (sebelum menggunakan aplikasi) dengan Post-Test (setelah penggunaan aplikasi).

b. Menentukan tingkat signifikansi

Pengujian menggunakan uji dua sisi dengan tingkat signifikansi α : 5 %. Tingkat signifikansi dalam hal ini berarti kita mengambil risiko salah dalam mengambil keputusan untuk menolak hipotesis yang benar sebanyak-banyaknya 5 % (signifikansi 5 % atau 0,05 adalah ukuran standar yang sering digunakan dalam penelitian).

c. Menentukan t hitung

Dapat dilihat dari hasil analisis uji *T Test*, dalam hal ini menggunakan software *SPSS 15* untuk menganalisis Uji *T Test* tersebut.

d. Menentukan T Tabel

Tabel distribusi t dicari pada $\alpha=5\% : 2 = 2,5\%$ (uji 2 sisi) dengan derajat kebebasan (df) $n-1$.

e. Kriteria Pengujian

H_0 diterima jika $-t \text{ tabel} \leq t \text{ hitung} \leq t \text{ tabel}$

H_0 ditolak jika $-t \text{ hitung} < -t \text{ tabel}$ atau $t \text{ hitung} > t \text{ tabel}$

Berdasar probabilitas :

H_0 diterima jika $P \text{ value} > 0,05$

H_0 ditolak jika $P \text{ value} < 0,05$

f. Membandingkan t hitung dengan t tabel dan probabilitas

Dari hasil perhitungan tersebut, maka t hitung akan dibandingkan dengan t tabel. Jika perbedaannya signifikan, maka disimpulkan bahwa hipotesa diterima. perhitungan t-test dengan lebih mudah dan cepat tanpa perlu melakukan perhitungan.

3.5 Antar Muka Pengguna

Graphical User Interface (GUI) merupakan antar muka pengguna suatu program berbasis grafis, yakni perintah-perintah tidak diketik melalui *key board*, berikut adalah beberapa tampilan antar pengguna untuk berinteraksi dengan sistem.

a. Form Awal

Gambar 8. Form Awal Pengukuran kualitas

b. Form Profil

Gambar 9. Form Profil Proyek Baru

c. Form Konfigurasi Bobot

Karakteristik	Bobot (%)
Kepercayaan Pengguna Software (versi baru)	11
1. Fungsionalitas	11
2. Keandalan	11
3. Keamanan	11
4. Efisiensi	11
5. Mudah Dipelajari	11
6. Portabilitas	11
7. Efisiensi	11
8. Mudah Dipelajari	11
9. Keamanan	11
10. Total	11

Gambar 10. Form Konfigurasi Bobot

d. Form Laporan

Gambar 11. Form Untuk Menampilkan Laporan

4. PEMBAHASAN

4.1 Instrumen Penilaian Implementasi

Faktor-faktor yang digunakan sebagai parameter untuk mengukur mempercepat aktivitas berbagi pengetahuan di perusahaan adalah partisipasi, motivasi, kemudahan, dan waktu. Dengan memperhatikan aspek-aspek tersebut, maka disusunlah item kuesioner sebagai berikut.

- Apakah menurut Anda saat ini, mengevaluasi perangkat lunak secara berkala mudah ditelusuri ?
- Apakah menurut Anda internal audit untuk mengevaluasi perangkat lunak saat ini lebih akurat ?

- Apakah menurut Anda melakukan evaluasi atau internal audit lebih mudah?
- Apakah menurut Anda, internal audit saat ini Anda lebih cepat dalam penyajian informasi?
- Apakah menurut Anda lebih mudah membuat laporan internal audit saat ini?
- Apakah menurut Anda saat ini lebih mudah dalam melakukan pertanyaan-pertanyaan?
- Apakah menurut Anda lebih melakukan penilaian khususnya untuk membuat *rating level*?
- Apakah menurut Anda akan mempergunakan lagi Aplikasi untuk melakukan evaluasi?

Pertanyaan-pertanyaan tersebut diukur dengan skala Likert. Skala Likert merupakan bentuk skala penilaian antara 1 (satu) sampai 4 (empat) dengan deskripsi sebagai berikut:

- Angka 1 (satu) menyatakan sangat tidak setuju
- Angka 2 (dua) menyatakan tidak setuju
- Angka 3 (tiga) menyatakan setuju
- Angka 4 (empat) menyatakan sangat setuju

Skala ini berfungsi untuk menunjukkan tingkat kepuasan responden terhadap hal yang ditanyakan.

4.2 Pengujian Aplikasi

Aplikasi *wizard* yang telah dibuat, selanjutnya diuji melalui teknik pengujian perangkat lunak yang meliputi pengujian *white box*.

a. Pengujian Aplikasi Pengujian *White Box*

Metode *white box* ini adalah suatu metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white box*, perikayasa sistem dapat melakukan *test case* yang dapat:

- Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
- Menggunakan semua keputusan logis pada sisi *true* dan *false*.
- Mengeksekusi semua *loop* (perulangan) pada batasan mereka dan pada batas operasional mereka

Dalam hal ini, pengujian tidak dilakukan terhadap keseluruhan program secara utuh, namun dilakukan sampel pengujian terhadap *proses* tertentu.

Pengujian *white Box* terhadap proses konfigurasi bobot, secara garis besar, algoritma dari proses login adalah sebagai berikut.

- masukan nilai karakteristik
- jika jumlah mencapai 100 %
- bila terjadi kurang atau lebih dari 100 persen dari total tampilkan pesan kesalahan
- lakukan pengisian data baru seperti langkah (a)
- bila telah mencapai total 100 %, maka user dapat mengakses sistem.

b. Pengujian Aplikasi Pengujian *Black Box*

Pengujian selanjutnya dilakukan untuk memastikan bahwa suatu *event* atau masukan akan menjalankan proses yang tepat dan menghasilkan output sesuai dengan rancangan. Untuk contoh pengujian terhadap beberapa perintah sebagai berikut :

Tabel 3. Hasil pengujian *Black Box*

Input	Proses	Output	Hasil Pengujian
Mencan tombol proyek baru	TotoPrngBheli_Click	Muncul layout Profile	Selesai
Menginput nama user	IdUserBheli_Click	Nama User	Selesai
Menginput kode	IdKategori_Click	Nama Kode	Selesai
Menginput tanggal audit	IdTglAudit_Click	Tanggal audit	Selesai
Mencan tombol next sampai	btNext_Click	Masuk ke frame konfigurasi bobot	Selesai
Mencan tombol kembali	btBack_Click	Kembali ke form semula	Selesai
Mencan tombol simpan laporan	IdFormTotoPrngBheli_Click	Mentampilkan laporan	Selesai

Kuesioner tersebut diisi oleh staf sebanyak 5 orang responden yang berhubungan langsung dengan kegiatan Internal Audit. Kuesioner dibagikan dua kali yaitu pada saat sebelum menggunakan Aplikasi Perangkat Lunak, serta setelah diimplementasikannya Aplikasi Perangkat Lunak.

Hasil nilai dari kuesioner yang diberikan pada pre-test adalah sebagai berikut:

Tabel 4. Hasil Pre-Test

No.	Nama	Pertanyaan								Total
		1	2	3	4	5	6	7	8	
1	Yoseph Anto	2	2	2	1	1	2	1	1	12
2	Rini Marini	1	2	1	2	1	2	1	1	11
3	Agus Sukoco	1	1	2	1	2	2	1	2	12
4	Yts. Angga	1	1	1	1	1	1	1	1	8
5	Tati	1	1	1	1	1	1	1	1	8

Setelah diimplementasikan sistem selanjutnya disebar kuisioner tahap kedua untuk mengukur dampak implementasi sistem, hasil nilai dari kuisioner yang diberikan adalah sebagai berikut:

Tabel 5. Hasil Post-Test

No.	Nama	Pertanyaan								Total
		1	2	3	4	5	6	7	8	
1	Yoseph Anto	3	2	3	2	3	3	3	3	17
2	Rini Marini	4	3	1	3	3	2	3	3	15
3	Agus Sukoco	3	3	1	2	3	2	4	3	13
4	Yts. Angga	3	4	1	2	2	3	2	3	11
5	Tati	3	4	1	3	2	3	3	3	13

Dari data hasil *pre test* dan *post test* yang telah dilakukan sebelumnya, dapat diringkas menjadi tabel berikut ini:

Tabel 6. Hasil Pre-Test dan Post-Test

No.	Nama	Pre-Test	Post-Test
1	Yoseph Anto	12	22
2	Rini Marini	11	23
3	Agus Sukoco	12	23
4	Yts. Angga	8	21
5	Tati	8	23

Data tersebut adalah hasil rangkuman dari hasil kuisioner yang sudah dijelaskan pada bab sebelumnya. Dari data tersebut, lalu data di-generate menggunakan fungsi *data analysis* dari spss 15 dan akan menghasilkan tabel berikut:

Tabel 7. Hasil T-Test

T-TEST
PAIRS = Sebelum, Setelah (PAIRED)
/CRITERIA = (1).95
/MISSING = ANALYSIS.

T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Paired	Sebelum Kuisioner	10,2000	5	2,04339	,91662
1	Setelah Kuisioner	22,4000	5	,68443	,40000

Paired Samples Correlations

		N	Correlation	Sig.
Paired	Sebelum Kuisioner & Setelah Kuisioner	5	,355	,208

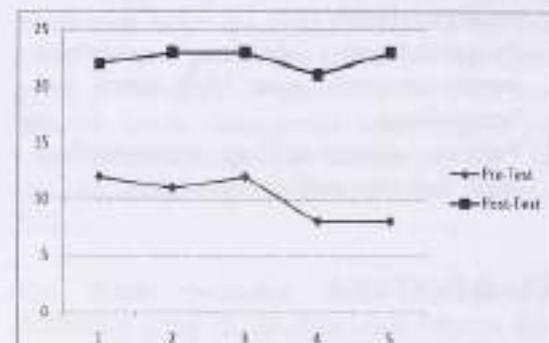
		Paired Samples Test				t	df	Sig. (2-tailed)	
		Paired Differences		95% Confidence Interval of the Difference					
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Paired	Sebelum Kuisioner & Setelah Kuisioner	12,20000	1,92054	,80025	14,30058	10,09942	14,182	4	,000

Dari tabel T-Test dapat dianalisis sebagai berikut:

- Dari hasil analisis *Paired samples statistics* dapat dilihat bahwa variabel responden jumlah data (N) sebanyak 5 dengan rata-rata 10,2 dan 22,4. Sedangkan standard deviasinya sebesar 2,0 (*pre test*) dan 0,8 (*post test*).
- Dari hasil analisis korelasi yang digunakan untuk mengetahui hubungan antara dua variabel dan untuk mengetahui arah hubungan yang terjadi. Maka korelasi yang terjadi adalah 0,355.
- T hitung dari tabel di atas didapat nilai t hitung adalah -14,182
- Tabel distribusi t dengan $\alpha = 5\%$ adalah 2,5% dengan derajat kebebasan (df) berdasarkan tabel di atas adalah 4. Dengan penujian 2 sisi (signifikansi = 0,025) hasil diperoleh untuk t tabel sebesar 2,776 (lihat lampiran tabel t).
- Nilai -t hitung < -t tabel -14 < -2,776 dan *P value* (0,000 < 0,05) maka H_0 ditolak.

Oleh karena nilai -t hitung < -t tabel dan *Pvalue* < ,05 maka H_0 ditolak, artinya bahwa ada perbedaan yang signifikan antara sebelum dan setelah penerapan aplikasi wizard efektif.

Apabila tabel hasil *pre test* dan *post test* digambarkan menjadi grafik, maka akan tampak perbedaan hasil dari sebelum dan sesudah penerapan aplikasi wizard untuk mengevaluasi perangkat lunak.



Gambar 8. Pre-Test dan Post-Test

Dari grafik tersebut, terlihat bahwa terjadi perbedaan signifikan terhadap hampir keseluruhan karyawan. Namun terlihat juga perbedaan yang sangat tipis pada beberapa karyawan. Hal ini disebabkan karena pada karyawan tersebut sebelumnya telah terbiasa mengaplikasikan mind map dan berbagi pengetahuan dengan menggunakan alat bantu.

5. SIMPULAN DAN SARAN

5.1 Simpulan

Dari hasil penelitian yang dilakukan mulai dari tahap awal hingga proses pengujian, dapat disimpulkan bahwa dengan diimplementasikannya aplikasi wizard untuk pengukuran perangkat lunak di perusahaan yang menjadi objek penelitian maka dapat disimpulkan sebagai berikut:

- Standard acuan yang digunakan untuk evaluasi perangkat dalam hal ini adalah standard ISO 9126 dapat digunakan untuk mengevaluasi perangkat lunak di PT. SMI
- Aplikasi wizard untuk mengukur evaluasi perangkat lunak dapat dipergunakan lebih efektif dan mempermudah pekerjaan terutama untuk mengevaluasi aplikasi penjualan di PT. SMI

5.2 Saran

Adapun saran-saran yang dapat penulis berikan adalah sebagai berikut:

- Perlu dikembangkan dengan versi berikutnya.
- Pengembangan integrasi dengan audit internal
- Pengembangan standard untuk bobot perangkat lunak
- Pengembangan dengan *customize* pertanyaan audit agar lebih detail dan berkembang.
- Perlunya dibuat aplikasi menggunakan lebih dari satu auditor dan auditee.

DAFTAR PUSTAKA

Bee Bee Chua, L. E. (2004). Applying the ISO 9126 model to the evaluation of an elearning. *ASCILITE*, 184-190.

- ISO. (2008). *Standard ISO 9001:2008*. Geneva: ISO Standard.
- ISO/IEC. (9001). *ISO/IEC 9126 - Software Product Evaluation-Quality Characteristics and Guidelines for their Use*. Geneva: ISO Standard.
- Jiménez-Pérez, G. (n.d.). *Design Wizards for Software Product Lines*. *Centro de Investigación en Informática*.
- Munawar. (2005). *Permodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.
- Myatt, A. (2008). *Pro NetBeans IDE 6*. New York: Apress.
- Nugroho, A. (2005). *Rationale Rose untuk Permodelan Berorientasi Objek*. Bandung: Penerbit Informatika.
- Penyusun, T. (2009). *Buku Pedoman Penulisan Proposal dan Tesis*. Jakarta: Eresha Education Indonesia.
- Pressman, R. S. (2003). *Rekayasa Perangkat Lunak*. Yogyakarta: Penerbit ANDI.
- Sutarto, R. H. (2009). *Konsep pemrograman JAVA dan Penerapannya untuk membuat software aplikasi*. Jakarta: Gramedia.
- Wahono, R. S. (2006). *Pengukuran Kualitas Perangkat Lunak*. *Majalah SDA*.
- Priyatno, D. (2009). *Mandiri Belajar SPSS untuk Analisis Data dan Uji Statistik*.