

**ANALISIS PENGEMBANGAN SISTEM APLIKASI *E-TRAINING*
KARYAWAN BERBASIS *WEB* PADA PT. MUTIARA SOLUSINDO**

SKRIPSI

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
guna Memperoleh Gelar Sarjana Pendidikan



Oleh

Imanaji Hari Sayekti

NIM 09520244011

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

APRIL 2013

**ANALISIS PENGEMBANGAN SISTEM APLIKASI *E-TRAINING*
KARYAWAN BERBASIS *WEB* PADA PT. MUTIARA SOLUSINDO**

SKRIPSI

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
guna Memperoleh Gelar Sarjana Pendidikan



Oleh

Imanaji Hari Sayekti

NIM 09520244011

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

APRIL 2013

PERSETUJUAN

Skripsi yang berjudul “ANALISIS PENGEMBANGAN SISTEM APLIKASI *E-TRAINING* KARYAWAN BERBASIS *WEB* PADA PT. MUTIARA SOLUSINDO” yang disusun oleh Imanaji Hari Sayekti, NIM 09520244011 ini telah disetujui oleh pembimbing untuk diujikan.

Yogyakarta, April 2013

Pembimbing,



Handaru Jati, Ph.D.

NIP. 19740511 199903 1 002

SURAT PERNYATAAN


Dengan ini saya menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Tanda tangan dosen penguji yang tertera dalam halaman pengesahan adalah asli. Jika tidak asli, saya siap menerima sanksi ditunda yudisium pada periode berikutnya.



Yogyakarta, 22 April 2013

Yang menyatakan,


Imanaji Hari Sayekti
NIM 09520244011

PENGESAHAN

Skripsi yang berjudul “ANALISIS PENGEMBANGAN SISTEM APLIKASI *E-TRAINING* KARYAWAN BERBASIS *WEB* PADA PT. MUTIARA SOLUSINDO” yang disusun oleh Imanaji Hari Sayekti, NIM 09520244011 ini telah dipertahankan di depan Dewan Penguji pada tanggal 22 April 2013 dan dinyatakan lulus.

DEWAN PENGUJI

Nama	Jabatan	Tanda Tangan	Tanggal
Handaru Jati, Ph.D	Ketua Penguji		25/4/13
Drs. Muhammad Munir M.Pd	Sekretaris Penguji		25/4/13
Dessy Irmawati, S.T.,MT	Penguji Utama		25/4/13

Yogyakarta, 22 April 2013

Universitas Negeri Yogyakarta

Fakultas Teknik

Dekan,




Dr. Mochamad Bruri Triyono, M.Pd.

NIP. 19560216 198603 1 003

MOTTO

“Asal Ada Kemauan Pasti Ada Jalan”

“Jangan Pernah Menyerah, Jangan Pernah Menyesal”

"Barang siapa yang bertakwa kepada Allah maka Allah akan memberikan jalan keluar baginya dari segala macam kesulitan dan Allah akan memberikan rezeki dari jalan yang tidak di sangka -sangjanya." (At - talaq :2)

"Allah menghendaki kemudahan kepadamu dan bukan menghendaki kesukaran." (Al-Bakarah: 185)

“Manusia tidak dibebani kecuali sesuai dengan batas kemampuannya” (QS. Al-Baqarah: 286).

PERSEMBAHAN

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga penulis bisa menyelesaikan skripsi ini.

Skripsi ini dipersembahkan untuk:

Kedua orang tua yakni Bapak dan Ibu yang senantiasa memberikan kasih sayang dan juga doanya.

Kemudian terima kasih juga atas segala sesuatu yang telah diberikan selama ini baik berupa materi maupun rohani sehingga penulis bisa seperti sekarang ini.

Semoga Allah membalas kebaikan Bapak dan Ibu

Amiin.

ANALISIS PENGEMBANGAN SISTEM APLIKASI *E-TRAINING* KARYAWAN BERBASIS *WEB* PADA PT. MUTIARA SOLUSINDO

Oleh
Imanaji Hari Sayekti
NIM 09520244011

ABSTRAK

Penelitian ini bertujuan untuk mengetahui analisis kualitas dari perangkat lunak sistem aplikasi *e-training* karyawan yang dibuat di PT. Mutiara Solusindo. Empat aspek yang menyangkut kualitas perangkat lunak diajukan untuk mengukur seberapa baik kualitas perangkat lunak yang dibuat.

Subjek penelitian ini adalah para karyawan dan aplikasi itu sendiri. Objek penelitian adalah aspek *reliability*, *usability*, *maintainability*, dan *portability*. Setting penelitian mengambil tempat di PT. Mutiara Solusindo dan perusahaan lain sebagai pengguna aplikasi. Metode yang digunakan dalam pengumpulan data adalah pengamatan langsung terhadap aplikasi *e-training* dan melakukan survei dengan menyebarkan angket kuisisioner yang dibuat ke dalam sebuah program aplikasi. Adapun metode yang digunakan untuk analisis data antara lain pengujian keandalan, analisis data skala likert, dan analisis langsung terhadap baris kode serta analisis kemampuan aplikasi dijalankan pada lingkungan yang berbeda.

Hasil penelitian menunjukkan bahwa kualitas dari sistem aplikasi *e-training* karyawan dilihat dari empat aspek, yaitu: (1) *reliability*, (2) *usability*, (3) *maintainability*, (4) *portability*. Secara keseluruhan telah memiliki kualitas yang baik sesuai dengan kaidah *software quality* dan harapan pengguna. Tingkat *reliability* tergolong baik, *usability* memiliki nilai persentase sebesar 81,44%, *maintainability index* berada pada angka 20-100, dan untuk aspek *portability* berdasarkan uji coba sistem dapat berjalan pada semua *browser* yang biasa digunakan pengguna.

Kata kunci: *kualitas perangkat lunak, e-training*

KATA PENGANTAR



Puji syukur penulis panjatkan kehadiran Allah SWT atas segala rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan tugas akhir Skripsi dengan judul “Analisis Pengembangan Sistem Aplikasi *E-Training* Karyawan Berbasis *Web* pada PT. Mutiara Solusindo” ini dengan baik dan lancar. Keberhasilan penulis dalam menyusun laporan ini tidak terlepas dari bantuan beberapa pihak, untuk itu penulis tidak lupa mengucapkan banyak terima kasih kepada :

1. Allah SWT yang atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir Akripsi ini.
2. Bapak dan Ibu yang tidak pernah kering akan do’a dan dukungannya selama penulis menyelesaikan Tugas Akhir Akripsi ini.
3. Prof. Dr. Rochmat Wahab, M.Pd, M.A selaku Rektor Universitas Negeri Yogyakarta.
4. Bapak Dr. Mochamad Bruri Triyono, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
5. Bapak Muhammad Munir, M.Pd. selaku Ketua Jurusan Pendidikan Teknik Elektronika, Fakultas Teknik Universitas Negeri Yogyakarta.
6. Ibu Dr. Ratna Wardani selaku Dosen Pembimbing Akademik.
7. Bapak Handaru Jati, Ph.D. selaku Dosen Pembimbing Tugas Akhir Skripsi.

8. Seluruh staf pengajar dan karyawan di Program Pendidikan Teknik Informatika, Fakultas Teknik Universitas Negeri Yogyakarta.
9. Bapak Hendrik Braja selaku HR Direktur PT. Mutiara Solusindo, Jakarta.
10. Bapak Acmad Haryadi selaku Pembimbing Penelitian di PT. Mutiara Solusindo, Jakarta.
11. Handwi Prasetyo selaku programmer yang telah memberi banyak sekali masukan, informasi dan bantuan selama penelitian.
12. Seluruh staf dan karyawan PT. Mutiara Solusindo, Jakarta.
13. Asti Wijayanti atas segala doa dan bantuannya.
14. Semua pihak yang telah membantu menyelesaikan laporan ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa Laporan Tugas Akhir Skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik yang bersifat membangun. Semoga apa yang penulis hasilkan dalam skripsi ini dapat bermanfaat bagi dunia pendidikan dan dapat menjadi referensi bagi teman-teman mahasiswa Universitas Negeri Yogyakarta.

Yogyakarta, 22 April 2013

Penulis

DAFTAR ISI

	Halaman
HAMAN JUDUL	i
PERSETUJUAN	ii
SURAT PERNYATAAN.....	iii
PENGESAHAN	iv
MOTTO.....	v
PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xv
BAB I_PENDAHULUAN	1
A. Latar Belakang.....	1
B. Identifikasi Masalah	6
C. Pembatasan Masalah.....	7
D. Perumusan Masalah.....	8
E. Tujuan Penelitian.....	8
F. Manfaat Penelitian.....	9
BAB II_KAJIAN PUSTAKA	11
A. Deskripsi Teori	11
1. Perancangan dengan UML.....	11
2. <i>Software Quality</i>	14
B. Kerangka Berpikir	24
C. Hipotesis	25
BAB III_METODE PENELITIAN.....	27

A. Desain Penelitian	27
1. Desain Penelitian Aspek <i>Reliability</i>	28
2. Desain Penelitian Aspek <i>Usability</i>	29
3. Desain Penelitian Aspek <i>Maintainability</i>	29
4. Desain Penelitian Aspek <i>Portability</i>	30
B. Populasi dan sampel	30
1. Populasi.....	30
2. Sampel.....	30
C. Teknik Pengumpulan Data	31
1. Data Primer	31
2. Data Sekunder.....	32
D. Instrumen Penelitian	33
E. Teknik Analisis Data	33
1. Analisis Data Aspek <i>Reliability</i>	33
2. Analisis Aspek <i>Usability</i>	34
3. Analisis Aspek <i>Maintainability</i>	35
4. Analisis Aspek <i>Portability</i>	36
BAB IV_HASIL PENELITIAN DAN PEMBAHASAN	37
A. Hasil Penelitian.....	37
1. Deskripsi Data.....	37
2. Deskripsi Sistem Aplikasi.....	37
B. Pembahasan	50
1. <i>Reliability</i>	50
2. <i>Usability</i>	54
3. <i>Maintainability</i>	57

4. <i>Portability</i>	59
BAB V_KESIMPULAN DAN SARAN.....	63
A. Kesimpulan.....	63
B. Saran	64
DAFTAR PUSTAKA	66

DAFTAR TABEL

	Halaman
Tabel 1. Karakteristik Kualitas McCall dan ISO 9126	2
Tabel 2. Tabel <i>Use Case Diagram</i>	12
Tabel 3. Tabel <i>Activity Diagram</i>	14
Tabel 4. Kategori <i>Maintainability</i>	36
Tabel 5. Definisi Aktor	38
Tabel 6. Definisi <i>Use Case</i>	39
Tabel 7. Skenario <i>Use Case Login</i>	40
Tabel 8. Skenario <i>Use Case Mastering Training</i>	41
Tabel 9. Skenario <i>Use Case Training Plan</i>	42
Tabel 10. Skenario <i>Use Case Implementing Training</i>	43
Tabel 11. Skenario <i>Use Case Evaluating Training</i>	44
Tabel 12. Skenario <i>Use Case View Training Record</i>	45
Tabel 13. Daftar Kesalahan dan Perbaikan	52
Tabel 14. Kategori Nilai <i>Availability</i>	53
Tabel 15. Hasil Pengujian Security	53
Tabel 16. Hasil Jawaban Kuisisioner	54
Tabel 17. Kategorisasi Kecenderungan Skor	56
Tabel 18. Pengujian <i>Portability Browser & Resolusi 1</i>	59
Tabel 20. Pengujian <i>Portability Browser & Device</i>	61
Tabel 21. Pengujian <i>Portability Device & Resolusi</i>	62

DAFTAR GAMBAR

	Halaman
Gambar 1. Faktor Kualitas McCall	16
Gambar 2. Kerangka Berpikir	24
Gambar 3. Diagram SDLC.....	27
Gambar 4. Tingkatan Persentase.....	35
Gambar 5. <i>Use Case Modul Training</i>	38
Gambar 6. <i>Login For All User</i>	45
Gambar 7. <i>Activity Diagram Preparing Master Training</i>	46
Gambar 8. <i>Activity Diagram Training Plan</i>	47
Gambar 9. <i>Activity Diagram Implementing Training</i>	47
Gambar 10. <i>Activity Diagram Evaluating Training</i>	48
Gambar 11. <i>Activity Diagram View Training Record</i>	48
Gambar 12. Tabel Database	49
Gambar 13. Hasil Uji <i>Stress Testing</i>	50
Gambar 14. Tingkatan Persentase.....	55
Gambar 15. MI Model.....	57
Gambar 16. MI Data Layer	57
Gambar 17. MI Business Layer	58

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Kuisisioner <i>Usability</i>	69
Lampiran 2. Development Record	71
Lampiran 3. Data Responden.....	73
Lampiran 4. User Acceptance Test (UAT)	75
Lampiran 5. Surat Keputusan Pembimbing	85
Lampiran 6. Keterangan Selesai Penelitian	87

BAB I

PENDAHULUAN

A. Latar Belakang

Minat terhadap kegiatan *training* karyawan di Asia Pasifik pada tahun 2012 dilaporkan meningkat. Khusus di Indonesia terjadi peningkatan sebesar 77% (Chandra, 2012). Peningkatan sebesar 77% tersebut tetap tidak dapat menutupi kenyataan bahwa 1 dari 10 karyawan yang mencari informasi pelatihan mencarinya melalui *Human Resource Department* (HRD), sedangkan sisanya mencari sumbernya sendiri. Di satu sisi hal ini menunjukkan minat besar terhadap pengembangan diri karyawan. Di sisi lain juga menunjukkan ketidakmampuan perusahaan dalam menyediakan program pengembangan diri.

Anggapan masa lalu bahwa kegiatan *training* karyawan tidak dapat meningkatkan nilai dan kesuksesan dalam menghadapi tantangan, saat ini telah berubah. Perusahaan yang menerapkan *training* terhadap karyawannya melaporkan kinerja keuangan yang lebih baik dibandingkan dengan kompetitor lain yang tidak menerapkan hal serupa. Meningkatnya minat karyawan juga mengharuskan pihak perusahaan memberi solusi agar *training* yang ada dapat terorganisasi dengan baik. Salah satu solusinya adalah dengan memanfaatkan teknologi. *Training* yang dahulu masih bersifat konvensional, sekarang telah dikemas dalam bentuk sistem aplikasi yang disebut *e-training*. Sistem aplikasi ini dapat mempermudah baik karyawan sebagai pihak yang membutuhkan *training* maupun para manajer untuk melakukan supervisi terhadap karyawannya.

Sistem aplikasi *e-training* dalam penggunaannya juga perlu diperhatikan, apakah sistem yang dibuat benar-benar dapat memenuhi kebutuhan penggunaanya atau malah sebaliknya. Apabila dalam proses pengembangan sistem aplikasi tidak sesuai dengan kaidah yang ada, maka kualitas dari sistem aplikasi tersebut juga akan sulit bisa dikatakan baik. Agar dapat mengetahui kualitas sistem aplikasi yang dibuat tersebut maka diperlukan suatu nilai yang dapat diukur (Pressman, 2012). Sementara untuk dapat mengukur kualitas perangkat lunak perlu dilakukan analisis dengan metode-metode pengukuran perangkat lunak. McCall bersama rekan-rekannya (McCall, Richards, & Walters, 1977) menjelaskan bahwa terdapat 3 aspek penting dalam produk perangkat lunak, yaitu: karakteristik operasional, kemampuan melakukan perubahan, dan kemampuan beradaptasi dengan lingkungan baru. Dari 3 aspek tersebut dijabarkan ke dalam 11 faktor yang dapat digunakan untuk mengukur kualitas perangkat lunak (Tabel.1). Sementara ISO 9126 (ISO9126, 2004) hanya menjabarkan 6 kunci dari kualitas perangkat lunak seperti yang ada pada Tabel 1 berikut:

Tabel 1. Karakteristik Kualitas McCall dan ISO 9126

McCall		ISO 9126
Perspective	Factors	Atribut
Product Operation	Correctness	Functionality
	Reliability	Reliability
	Efficiency	Usability
	Integrity	Efficiency
	Usability	Maintainability
Product Revision	Maintainability	Portability
	Testability	
	Flexibility	
Product Transaction	Portability	
	Reusability	
	Interoperability	

Banyak peneliti mengusulkan akan metode pengukuran berdasarkan penjabaran-penjabaran tersebut. Tetapi pada kenyataannya setiap perangkat lunak juga memiliki karakteristiknya masing-masing. Seperti perangkat lunak yang berbasis web, menurut Gerti Kappel dan rekan-rekannya (Kappel, Proll, Reich, & Retschitzegger, 2006) menyatakan bahwa terdapat 6 faktor penting yang dapat menentukan kesuksesan perangkat lunak berbasis web. Faktor-faktor tersebut adalah, *usability*, *availability*, *browser compatibility*, *security*, *actuality*, dan *efficiency*. Sebuah kualitas perangkat lunak tidak dapat diukur hanya pada saat perangkat lunak tersebut selesai dibuat. Tetapi analisis pada sisi pengembangan juga perlu diperhatikan, karena tidak dapat dipungkiri bahwa dengan proses pengembangan yang baik, tentu akan berpengaruh terhadap hasil akhir produk yang dibuat.

Pada beberapa kasus permasalahan yang terjadi di PT. Mutiara Solusindo dan juga kliennya. Terdapat beberapa hal yang menjadi sorotan baik bagi pihak pengembang maupun pengguna (klien). Beberapa hal tersebut seperti, klien yang lebih memperhatikan bagaimana perangkat lunak yang telah dibuat memiliki keandalan yang baik, sehingga ketika dibutuhkan perangkat lunak tersebut tidak mengalami masalah dan dapat memenuhi kebutuhannya. Sistem aplikasi perangkat lunak sebagai sebuah produk dari PT.Mutiara Solusindo bagaimanapun juga memiliki karakteristik khusus yang memang tidak mudah dimengerti, sehingga dapat menimbulkan beberapa masalah ketika perangkat lunak tersebut digunakan. Seperti contohnya pada penggunaan perangkat lunak dalam jangka waktu tertentu mungkin akan ada banyak cacat sistem aplikasi yang belum

diketahui sampai akhirnya terdeteksi. Selama berbulan-bulan atau bahkan bertahun-tahun selama sistem perangkat lunak tersebut masih digunakan, tentu akan ada banyak lagi cacat-cacat lain yang belum ditemukan. Jim A. McCall bersama rekan-rekannya (McCall et al., 1977) membahas mengenai kegagalan dan kecacatan pada sistem aplikasi perangkat lunak pada sebuah kajian yang disebut *reliability*.

Di sisi lain dari kebutuhan pengguna akan keandalan sistem, pihak pengembang juga tidak menginginkan adanya kesulitan saat memperbaiki kesalahan-kesalahan yang terjadi dari perangkat lunak yang dibuatnya. Pihak pengembang menginginkan bahwa perangkat lunak yang dibuatnya dapat dengan mudah dilakukan pemeliharaan atau perubahan jika diperlukan. Sebuah sistem aplikasi yang dirancang dan dikembangkan dengan baik, tentu akan dapat dengan mudah dilakukan pemeliharaan sistem. Pemeliharaan sistem menurut Sommerville (Sommerville, 2003) dibagi menjadi 3 jenis, salah satunya yang terkait pemeliharaan yaitu untuk memperbaiki kesalahan. Seberapa jauh perangkat lunak dapat diperbaiki dari kesalahan atau dipelihara merupakan faktor yang masuk dalam kajian *maintainability* perangkat lunak menurut McCall dan rekan-rekannya (McCall et al., 1977).

Sebuah perangkat lunak akan mengalami perubahan atau penambahan tidak hanya dikarenakan adanya kesalahan yang ditemukan, tetapi juga karena unsur kepuasan dari pengguna dalam berinteraksi dengan perangkat lunak tersebut. Seorang pengguna dengan pengguna lain akan memiliki tingkat kepuasan yang berbeda-beda. Beberapa pengguna menginginkan tampilan-tampilan yang

memukau, beberapa pengguna lainnya hanya menginginkan tampilan dengan teks yang sederhana. Selain sebagai pengguna merekalah yang berperan menentukan kepuasan dari produk perangkat lunak yang telah dikembangkan. Kepuasan tersebut dapat diperoleh dari pengalaman mereka ketika berinteraksi dengan perangkat lunak yang telah dibuat. Tingkat kepuasan sendiri atau tanggapan dari para pengguna akan sangat berguna bagi pihak pengembang, karena dari tanggapan itulah seorang pengembang dapat melakukan koreksi terhadap aplikasi yang dibuatnya. Jakob Nielsen (Nielsen, 1993) menjelaskan mengenai pengukuran pengalaman pengguna dalam berinteraksi dengan sistem perangkat lunak ke dalam suatu pokok pembahasan yang dinamakan *usability*.

Kebutuhan masing-masing pengguna yang berbeda dalam berinteraksi dengan sistem juga dapat berimplikasi pada banyaknya pengguna yang fanatik atau terbiasa terhadap sistem operasi tertentu, aplikasi tertentu, bahkan perangkat tertentu. PT. Mutiara Solusindo selaku pengembang produk perangkat lunak dalam proses pengembangan aplikasinya selalu dihadapkan pada lingkungan Framework .NET dari Windows yang berbasis *web*. Bahasa pemrograman yang digunakan adalah bahasa pemrograman ASP.NET dan C#. Hal ini tentu akan menimbulkan suatu pertanyaan, apakah proses pengembangan yang menggunakan bahasa pemrograman berbasis windows dapat menghasilkan perangkat lunak yang mampu memenuhi kebutuhan kliennya. Seberapa mudahnya sebuah perangkat lunak dapat berjalan pada lingkungan sistem yang berbeda merupakan sebuah definisi dari *portability* perangkat lunak menurut McCall (McCall et al., 1977).

PT. Mutiara Solusindo sebagai salah satu perusahaan yang bergerak dalam bidang pengembangan perangkat lunak sangat memperhatikan peningkatan kualitas Sumber Daya Manusia (SDM) yang dimiliki perusahaan-perusahaan kliennya. Salah satu buktinya adalah, saat ini PT. Mutiara Solusindo telah banyak menghasilkan perangkat lunak yang terkait dengan peningkatan kualitas SDM perusahaan. Salah satu diantaranya adalah aplikasi *e-training* karyawan. Selain karena permintaan untuk pembuatan aplikasi *e-training* yang meningkat, di sisi lain juga karena kemauan para karyawan untuk mengembangkan diri juga meningkat. Beberapa alasan inilah yang menyebabkan PT.Mutiara Solusindo dipilih sebagai tempat penelitian oleh penulis.

Berdasarkan permasalahan-permasalahan yang telah dijabarkan di atas maka penulis tertarik melakukan penelitian dengan judul “Analisis Pengembangan Sistem Aplikasi *E-Training* Karyawan Berbasis *Web* pada PT. Mutiara Solusindo”

B. Identifikasi Masalah

Dari latar belakang yang dikemukakan di atas, dapat diidentifikasi masalah-masalah sebagai berikut:

1. Banyak perusahaan di Indonesia belum dapat menyediakan program pengembangan diri yang dibutuhkan karyawan dan manajernya.
2. Masih saja terdapat kesalahan yang ditemukan saat perangkat lunak sudah di tangan klien, sehingga dapat mengakibatkan pembengkakan biaya dan ketidakefisienan waktu.
3. Belum diketahui sejauh mana keandalan (*reliability*) perangkat lunak yang dibuat di PT. Mutiara Solusindo

4. Pihak pengembang tidak menginginkan adanya kesulitan saat memperbaiki kesalahan yang terjadi dari perangkat lunak yang dibuat.
5. Belum diketahui ukuran seberapa jauh atau mudahnya perangkat lunak yang dibuat untuk mengalami perubahan (*maintainability*).
6. Belum pernah diketahui sebelumnya mengenai pengalaman masing-masing pengguna dalam berinteraksi dengan perangkat lunak yang dibuat (*usability*).
7. Proses pengembangan aplikasi masih terbatas pada lingkungan windows.
8. Belum diketahui seberapa mudah aplikasi yang dikembangkan dapat dijalankan oleh para pengguna dengan lingkungan sistem bermacam-macam (*portability*).
9. Belum diketahui seberapa baik proses pengembangan sistem aplikasi *e-training* untuk dapat memenuhi kebutuhan penggunanya
10. PT. Mutiara Solusindo belum memiliki manajemen kualitas secara khusus untuk menangani aspek kualitas perangkat lunak yang dibuatnya

C. Pembatasan Masalah

Dengan pertimbangan permasalahan yang akan dihadapi menyangkut aspek-aspek pada *software quality* yang ada. Penulis membatasi beberapa aspek dari *software quality* yang dijadikan pedoman analisis pengembangan sebagai berikut:

1. Analisis aspek *reliability* sesuai dengan karakteristiknya sebagai aplikasi web.
2. Analisis aspek *usability* yang digunakan untuk mengukur pengalaman pengguna berinteraksi dengan sistem.
3. Analisis aspek *maintainability* yang berfokus pada indeks yang diperoleh.
4. Analisis aspek *portability* yang berfokus pada lingkungan pengguna akhir.

D. Perumusan Masalah

Berdasarkan identifikasi masalah di atas dirumuskanlah rumusan masalah yang akan diteliti yaitu sebagai berikut:

1. Bagaimana cara menganalisis aspek *reliability* pada sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo?
2. Bagaimana cara menganalisis aspek *usability* pada sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo?
3. Bagaimana cara menganalisis aspek *maintainability* pada sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo?
4. Bagaimana cara menganalisis aspek *portability* pada sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo?

E. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan penelitian ini adalah untuk:

1. Mengetahui analisis aspek *reliability* sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo sehingga aplikasi tersebut dapat digunakan oleh pengguna dengan baik tanpa ada masalah.
2. Mengetahui analisis aspek *usability* sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo sehingga dapat dijadikan media koreksi terhadap aplikasi yang dikembangkan.
3. Mengetahui analisis aspek *maintainability* sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo sehingga para *programmer* akan lebih berhati-hati dalam melakukan *coding*.

4. Mengetahui analisis aspek *portability* sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo sehingga dapat dibuktikan bahwa dengan proses pengembangan yang selalu berada pada lingkungan windows bukan berarti bahwa aplikasi yang dihasilkan tidak dapat digunakan oleh pengguna dengan latar belakang sistem dan perangkat yang berbeda.

F. Manfaat Penelitian

Dengan adanya penelitian ini diharapkan akan diperoleh beberapa manfaat baik bagi penulis, pengembang maupun pengguna sistem aplikasi *e-training* karyawan. Manfaat yang dapat diperoleh dari penelitian ini antara lain:

1. Manfaat Teoritis

Hasil penelitian ini dapat dijadikan referensi mengenai bagaimana melakukan uji kualitas perangkat lunak dilihat dari sisi *reliability*, *usability*, *maintainability* dan *portability* sesuai dengan teori-teori yang disampaikan oleh para ahli.

2. Manfaat Praktis

Manfaat yang dapat diperoleh dari sisi pengembang antara lain:

- a. Dapat diketahui hasil analisis pengembangan sistem aplikasi *e-training* dari aspek *reliability* sehingga dapat digunakan sebagai koreksi dan antisipasi terhadap kesalahan-kesalahan yang mungkin terjadi pada perangkat lunak yang dibuat.
- b. Diketahui tanggapan secara langsung dari setiap pengguna sistem aplikasi *e-training* karyawan, sehingga nantinya tanggapan tersebut dapat dijadikan koreksi terhadap aplikasi *e-training* yang dikembangkan.

- c. Dapat diketahui indeks perawatan pada sistem aplikasi *e-training* karyawan yang nantinya dapat dipergunakan sebagai bahan evaluasi untuk semakin meningkatkan kualitas aplikasi yang dikembangkan.
- d. Diketahui analisis portabilitas sistem aplikasi *e-training* terhadap kebutuhan-kebutuhan pengguna dengan latar belakang sistem dan perangkat yang berbeda-beda.
- e. Dapat dijadikan parameter pengujian untuk pengembangan aplikasi-aplikasi lain yang ada di PT.Mutiara Solusindo selain aplikasi *e-training* karyawan.

Sedangkan dari sisi pengguna antara lain:

- a. Pengguna dapat menyampaikan masukannya secara langsung pada pengembang dengan adanya *questionnaire usability*.
- b. Pengguna semakin dipermudah karena sistem aplikasi yang dibuat dapat diakses melalui berbagai perangkat *mobile* maupun *desktop*.

BAB II KAJIAN PUSTAKA

A. Deskripsi Teori

1. Perancangan dengan UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam dunia industri untuk mendefinisikan kebutuhan, membuat analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek (S. & Shalahudin, 2011). UML menawarkan sebuah standar untuk merancang model sistem. UML dapat diterapkan oleh seorang pengembang perangkat lunak untuk membuat model pada berbagai jenis aplikasi, baik menyangkut lingkungan perangkat keras yang bermacam-macam maupun pada berbagai bahasa pemrograman. Di dalam UML terdapat beberapa konsep dasar terkait komponen-komponen diagram seperti:

a. *Use Case Diagram*




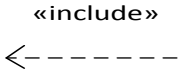

Use case diagram merupakan suatu cara pemodelan untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem (S. & Shalahudin, 2011). *Use case diagram* mendeskripsikan sebuah interaksi antara satu aktor atau lebih dengan sebuah sistem yang akan dibuat. Dalam artian bahwa sebuah *use case diagram* dapat digunakan untuk menggambarkan fungsi-fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang menggunakan fungsi-fungsi tersebut. Dalam pembuatan sebuah *use case diagram* yang perlu ditekankan adalah “apa” yang diperbuat sistem, bukan “bagaimana” sebuah sistem berjalan. Dalam pendefinisian sebuah nama juga perlu sesimpel mungkin namun tetap mudah

dipahami agar pembaca merasa nyaman ketika membaca *use case diagram*. Ada dua hal paling utama dalam sebuah *use case diagram* yaitu aktor dan *use case*.

1. Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat. Jadi meskipun simbol aktor berupa gambar orang, belum tentu yang dimaksud aktor adalah orang. Bisa saja sistem lain di luar sistem yang akan dibuat atau sebuah proses.
2. *Use case* merupakan fungsionalitas yang disediakan dalam sebuah sistem sebagai unit-unit yang dapat saling bertukar pesan dengan unit lainnya maupun dengan aktor. Penulisan nama pada sebuah *use case* biasa dinyatakan dalam sebuah kata kerja pada awal frasa.

Deskripsi dan simbol *use case diagram* disajikan dalam Tabel 2 berikut:

Tabel 2. Tabel *Use Case Diagram*

Simbol	Deskripsi
Use case 	Menggambarkan fungsionalitas sistem
Aktor 	Bisa orang, proses, atau sistem lain
Asosiasi 	Komunikator antara aktor dan <i>use case</i> yang berpartisipasi
Include 	Sebagai syarat dijalankannya sebuah <i>use case</i>
Generalisasi 	Menggeneralisasikan fungsi

b. *Activity Diagrams*

Activity diagrams adalah diagram yang menggambarkan berbagai aliran kerja atau aktivitas dari sebuah sistem (S. & Shalahudin, 2011). Dalam membuat sebuah diagram aktivitas yang perlu diperhatikan adalah bahwa sebuah *activity*


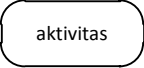



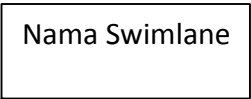
diagram adalah menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Jadi dengan kata lain aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas banyak digunakan untuk mendefinisikan hal-hal berikut:

- 1) Rancangan proses bisnis, yaitu digunakan untuk merancang urutan-urutan dari aktivitas sistem yang didefinisikan.
- 2) Rancangan pengujian, yaitu bahwa setiap aktivitas yang dianggap memerlukan sebuah pengujian perlu didefinisikan kasus ujinya.

Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Deskripsi dan simbol *activity diagram* disajikan dalam Tabel 3 berikut:

Tabel 3. Tabel *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem
Aktivitas 	Aktivitas yang dilakukan sistem
<i>Decision</i> 	Asosiasi percabangan apabila ada pilihan aktivitas lebih dari satu
<i>Fork/Join</i> 	Penggabungan lebih dari satu aktivitas
Status akhir 	Status akhir yang dilakukan sistem
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2. *Software Quality*

Menurut Ladjamudin (Ladjamudin, 2006) kualitas perangkat lunak adalah suatu kesesuaian terhadap kebutuhan fungsional maupun kinerja perangkat lunak secara eksplisit, standar pengembangan yang juga terdokumentasi secara eksplisit, dan karakteristik implisit yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional. Sedangkan menurut Roger S. Pressman (Pressman, 2012) kualitas perangkat lunak didefinisikan sebagai suatu proses dari perangkat lunak yang efektif diterapkan sehingga menyediakan nilai agar dapat diukur bagi kepentingan mereka para produsen maupun pengguna perangkat lunak.

Menurut Gerti Kappel dan rekan-rekannya (Kappel, Proll, Reich, & Retschitzegger, 2006) menyatakan bahwa terdapat 6 faktor penting yang dapat menentukan kesuksesan perangkat lunak berbasis web. Faktor-faktor tersebut adalah, *usability*, *availability*, *browser compatibility*, *security*, *actuality*, dan *efficiency*.. Kemudian, *International Standard Organization* melalui ISO-9126

(ISO9126, 2004) mengidentifikasi enam karakteristik dalam pengukuran kualitas perangkat lunak yaitu, *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, dan *portability*.

Berdasarkan definisi-definisi di atas ada tiga hal yang menjadi pokok terpenting dalam kajian kualitas perangkat lunak. Hal-hal pokok tersebut adalah sebagai berikut:

1. Memenuhi kebutuhan pengguna

Apabila perangkat lunak tidak dapat memenuhi kebutuhan penggunanya, maka perangkat lunak tersebut dikatakan tidak atau kurang memiliki kualitas baik.

2. Memenuhi standar pengembangan perangkat lunak

Artinya bahwa apabila dalam pengembangan perangkat lunak pihak pengembang tidak memperhatikan atau tidak mengikuti metodologi yang sudah menjadi dasar, maka hampir dipastikan kualitas dari perangkat lunak tersebut akan sulit bisa dikatakan baik.

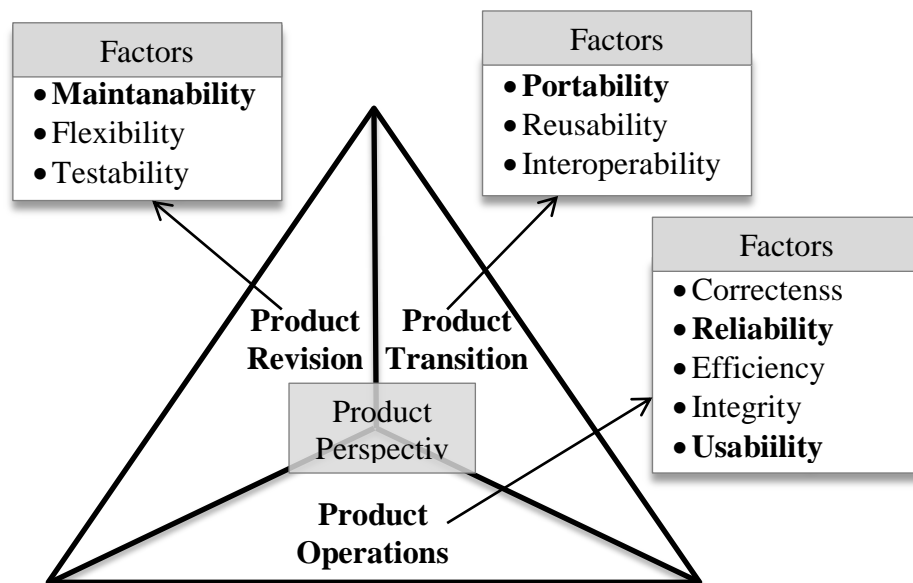
3. Memenuhi kriteria pokok yang mutlak diperlukan

Artinya apabila salah satu kriteria yang mutlak diperlukan dalam perangkat lunak tersebut tidak dapat terpenuhi, maka perangkat lunak tersebut tidak dapat dikatakan memenuhi kualitas yang baik.

Jim A. McCall dan rekan-rekannya (McCall et al., 1977) menyampaikan bahwa dalam pengembangan sebuah perangkat lunak terdapat tiga hal penting yang dapat mempengaruhi kualitas dari perangkat lunak. McCall menggolongkan tiga hal tersebut ke dalam sudut pandang pengembangan produk perangkat lunak

yang dikenal dengan sebutan *product perspective*. Tiga hal dari *product perspective* menurut McCall adalah sebagai berikut:

1. Sifat-sifat operasional perangkat lunak (*Product Operations*);
2. Kemampuan perangkat lunak untuk diubah (*Product Revision*);
3. Penyesuaian perangkat lunak terhadap lingkungan (*Product Transition*).



Gambar 1. Faktor Kualitas McCall

Dari Gambar 1 mengenai faktor kualitas McCall di atas kemudian dijelaskan definisi dari masing-masing *product perspective* yang ada sebagai berikut:

a. *Product Operation*

Product Operation menurut McCall berisi mengenai sifat-sifat operasional dari perangkat lunak. Sifat-sifat operasional ini adalah suatu hal yang perlu diketahui dan diperhatikan oleh para pengembang ketika mengembangkan sebuah aplikasi, karena tahap perancangan sebuah aplikasi akan masuk ke dalam lingkup

product operation (McCall et al., 1977). Faktor-faktor menurut McCall yang berkaitan dengan sifat-sifat operasional perangkat lunak dua diantaranya, yaitu:

1) *Reliability*

Merupakan aspek yang berkaitan dengan kemungkinan suatu perangkat lunak dapat melakukan fungsi-fungsi tertentu sesuai dengan tingkat ketelitian yang diinginkan (Pressman, 2012). Jadi faktor *reliability* merupakan sebuah parameter dari keandalan sistem dalam menjalankan fungsi-fungsinya sehingga dapat terhindar dari berbagai macam kegagalan sistem.

Kegagalan dalam perangkat lunak bisa berarti ketidaksesuaian dengan kebutuhan yang diharapkan. Beberapa kegagalan dalam perangkat lunak biasanya dapat diperbaiki hanya dalam waktu beberapa menit, namun dari perbaikan kegagalan tadi juga tidak menutup kemungkinan akan munculnya kesalahan baru, yang mungkin membutuhkan waktu perbaikan lebih lama lagi. Bahkan dengan ditemukannya kesalahan-kesalahan baru dari perbaikan yang pertama, setelah berhasil diperbaiki, hal ini tidak menutup kemungkinan lagi bahwa kesalahan-kesalahan lain yang seterusnya bisa terjadi.

Karena itu, diciptakanlah beberapa metode untuk melakukan analisis terhadap *reliability* perangkat lunak. Salah satunya (Tian, Ma, Li, & Koru, 2003) menjabarkan tiga strategi yang dapat ditempuh untuk menganalisis *reliability* khusus untuk aplikasi berbasis web. Tiga hal tersebut adalah sebagai berikut:

1. Membangun model pengujian statistik atau model penggunaan berdasarkan skenario dan frekuensinya.

2. Gunakan model statistik untuk uji kasus pengembangan sampai mencoba menjalankan aplikasi
3. Menganalisis hasil tes untuk penilaian keandalan dan prediksi untuk pengambilan keputusan.

Kemudian ada juga tambahan perhitungan yang terkait dengan *reliability* menurut Pressman yaitu *availability*. *Availability* adalah ketersediaan sistem setiap kali diperlukan pada waktu tertentu. (Pressman, 2012). Perhitungan yang terkait dengan *availability* dijabarkan seperti berikut:

$$Availability = \frac{MTTF}{(MTTF+MTTR)} \times 100\%$$

MTTF (*mean time to failure*) adalah waktu rata-rata terjadinya kesalahan dalam satu sistem, sedangkan MTTR (*mean time to repair*) adalah waktu rata-rata sistem tersebut mengalami perbaikan.

2) *Usability*

(Nielsen, 1993) mendefinisikan *usability* sebagai ukuran kualitas pengalaman pengguna ketika berinteraksi dengan perangkat lunak yang dibuat. *International Standard Organization* (ISO9126, 2004) juga mendefinisikan *usability* sebagai sesuatu yang berkaitan dengan bagaimana sebuah produk dapat digunakan oleh pengguna tertentu untuk dapat mencapai suatu tujuan secara efektif, efisien, dan memuaskan dari sisi konteks penggunaan. (McCall et al., 1977) mendefinisikan *usability* sebagai usaha yang diperlukan untuk mempelajari, menyiapkan *input*, dan menginterpretasikan *output* dari suatu program.

Istilah *usability* dalam pengembangan perangkat lunak menurut Jakob Nielsen (Nielsen, 1993) merujuk pada beberapa atribut yang dapat mempengaruhi

kemudahan pemakaian perangkat lunak dan atribut tersebut juga dapat dijadikan dasar pengukuran *usability*. Beberapa atribut tersebut terdiri dari lima, yaitu:

1. *Learnability*, dapat digunakan untuk mengukur tingkat kemudahan melakukan tugas sederhana pada pemakaian pertama.
2. *Efficiency*, dapat digunakan untuk mengukur kecepatan mengerjakan tugas tertentu setelah mempelajari desain.
3. *Memorability*, dapat digunakan untuk mengukur seberapa cepat pengguna mendapatkan kembali kecakapan dalam menggunakan desain tersebut setelah beberapa waktu tidak menggunakan.
4. *Errors*, dapat digunakan untuk mengukur seberapa banyak kesalahan yang dilakukan pengguna, seberapa apa kesalahan yang dibuat, dan seberapa apa mereka mendapatkan penyelesaian.
5. *Satisfaction*, dapat digunakan untuk mengukur tingkat kepuasan dalam menggunakan desain yang telah dibuat.

Kelima kriteria pengukuran tersebut akan dibahas dalam bentuk pertanyaan dan jawaban. Pertanyaan yang diajukan merupakan syarat suatu perangkat lunak memenuhi kriteria *usability* tertentu, sehingga jawabannya merupakan tolak ukur dari pemenuhan *usability*. Pertanyaan-pertanyaan tersebut merupakan bahan uji untuk mengetahui seberapa jauh kualitas *usability* perangkat lunak yang dibuat.

Pengujian dilakukan dengan melakukan survei terhadap pengguna sistem dengan menggunakan angket kuisisioner J.R Lewis yang telah dipublikasikan pada *International Journal of Human Computer Interaction* pada tahun 1995 (Lewis, Group, & Boca Raton, 1995).

b. *Product Revision*

Apabila sebuah perangkat lunak telah berhasil dikembangkan dan diimplementasikan, kemudian dilakukan uji coba maupun evaluasi. Dalam tahap ini akan muncul berbagai hal yang perlu diperbaiki. Seperti perbaikan yang menyangkut kemudahan pemeliharaan sampai memastikan perangkat lunak yang dikembangkan dapat melakukan fungsinya dengan baik (McCall et al., 1977).

Seberapa jauh perangkat lunak dapat diperbaiki atau mengalami perubahan merupakan faktor lain yang harus diperhatikan. McCall membagi beberapa faktor yang menyangkut kemampuan perangkat lunak dalam menjalani perubahan salah satunya adalah *maintainability*. *Maintainability* merupakan besarnya upaya yang diperlukan untuk melokalisasi dan memperbaiki kesalahan-kesalahan yang ada (Pressman, 2012). Perangkat lunak yang berhasil dikembangkan kemudian diimplementasikan pada kenyataannya akan ada saja hal-hal yang perlu diperbaiki atau ditambahkan. Sebuah sistem perangkat lunak yang dirancang dan dikembangkan dengan baik, tentu akan dengan mudah dapat dilakukan revisi atau penambahan jika diperlukan. Menurut (Sommerville, 2003) pemeliharaan perangkat lunak dibagi menjadi tiga jenis, yaitu:

1. Pemeliharaan untuk memperbaiki kesalahan perangkat lunak

Perbaikan pada *error coding* biasanya membutuhkan biaya yang relatif lebih murah dibandingkan *error* perancangan, karena *error* perancangan dapat melibatkan penulisan ulang beberapa komponen program. Bahkan biaya pengeluaran akan jauh lebih mahal lagi jika persyaratan yang mengalami *error*, karena *error* persyaratan membutuhkan perancangan ulang secara luas.

2. Pemeliharaan untuk menyesuaikan perangkat lunak pada lingkungan

Jenis pemeliharaan ini diperlukan ketika beberapa aspek lingkungan sistem seperti perangkat keras, perangkat lunak, maupun sistem operasi yang digunakan berubah. Apabila tidak direncanakan dari awal maka modifikasi terhadap sistem aplikasi adalah suatu keharusan agar sistem yang ada dapat berjalan pada lingkungan yang baru

3. Pemeliharaan untuk menambah atau memodifikasi fungsionalitas

Hal ini terjadi biasanya dikarenakan terjadinya perubahan kebijakan bisnis atau organisasi. Dalam hal ini juga perlu diantisipasi bahwa biasanya skala perubahan yang dibutuhkan bagi perangkat lunak dengan pemeliharaan jenis ini akan jauh lebih besar jika dibandingkan dengan jenis pemeliharaan lainnya.

Seberapa jauh dan mudahnya perangkat lunak untuk dimodifikasi dapat dinyatakan dalam sebuah nilai indeks yang disebut *Maintainability Index* (MI). *Maintainability index* adalah metrik perangkat lunak yang mengukur seberapa mudah suatu perangkat lunak dapat dirawat apabila suatu ketika diperlukan perubahan terhadap kode program yang ada.

Oman dan rekan-rekannya (Oman & Hagemester, 1992) mengemukakan mengenai hirarki pemeliharaan perangkat lunak. Di dalam hirarki tersebut disampaikan bahwa *Halstead Complexity* dan *Cyclomatic Complexity* merupakan indikator dari *maintainability*. (Coleman, Ash, Lowther, & Oman, 1994) juga mengemukakan bahwa *maintainability index* merupakan suatu pendekatan yang paling umum digunakan untuk mengukur *maintainability*.

Kemudian seiring dengan perkembangan teknologi saat ini perhitungan untuk mencari nilai *maintainability index* telah dikemas dalam bentuk perangkat lunak. Seperti contoh SEI (*Software Engineering Institute*) yang menggunakan rumus sebagai berikut dalam perhitungan nilai *maintainability index*.

$$MI = 171 - 5.2 * \log_2(V) - 0.23 * G - 16.2 * \log_2(LOC) + 50 * \sin(\sqrt{2.4 * CM})$$

Keterangan: V = *Halstead Volume*

G = *Cyclomatic Complexity*

LOC = *count of source Lines Of Code* (SLOC)

CM = *percent of lines of Comment* (optional)

Sedangkan pihak Microsoft melalui website resminya memaparkan rumus-rumus yang digunakan dalam perhitungan nilai *maintainability index* sama dengan yang disampaikan (Chawla & Chhabra, 2012) sebagai berikut.

$$MI = \text{MAX}(0, (171 - 5.2 * \ln(V) - 0.23 * (G) - 16.2 * \ln(LOC)) * 100 / 171)$$

Pihak Microsoft melalui website resminya memberikan pandangannya tentang nilai-nilai, ikon, dan warna yang ada pada hasil analisis kode dengan menggunakan *tools* analisis dari Visual Studio 2008. Dalam *library* dari Microsoft Development Network (MSDN) juga dijelaskan bahwa apabila nilai *maintainability index* berada pada angka 20-100 maka hal ini bisa mengindikasikan bahwa kode program tersebut mudah untuk dilakukan perawatan apabila pada suatu ketika diperlukan perubahan. Sementara apabila nilai *maintainability index* berada pada angka 10-19, ini bisa dikatakan kode program tersebut memiliki kemudahan pemeliharaan yang cukup. Namun apabila *maintainability index* berada pada angka 0-9, hal ini perlu diperhatikan karena

kode program yang dibuat akan sulit untuk dilakukan perawatan atau dengan kata lain kemudahan perawatan terhadap kode program rendah.

c. *Product Transition*

Setelah kualitas dari perangkat lunak diukur secara teknis dengan menggunakan faktor *product operation* dan secara implementasi telah disesuaikan dengan faktor *product revision*, faktor terakhir yang harus diperhatikan dalam proses pengembangan perangkat lunak adalah faktor *product transition* (McCall et al., 1977). Di dalam *product transition* terdapat faktor-faktor yang berisi tentang bagaimana sebuah perangkat lunak dapat dijalankan pada lingkungan platform yang berbeda atau kerangka sistem yang berlainan. Faktor-faktor McCall yang terkait dengan tingkat adaptabilitas perangkat lunak terhadap lingkungan baru adalah *portability*. *Portability* adalah aspek yang berkaitan dengan usaha yang diperlukan untuk dapat mentransfer sebuah program dari sebuah lingkungan perangkat keras atau lunak tertentu ke lingkungan yang lain (McCall et al., 1977).

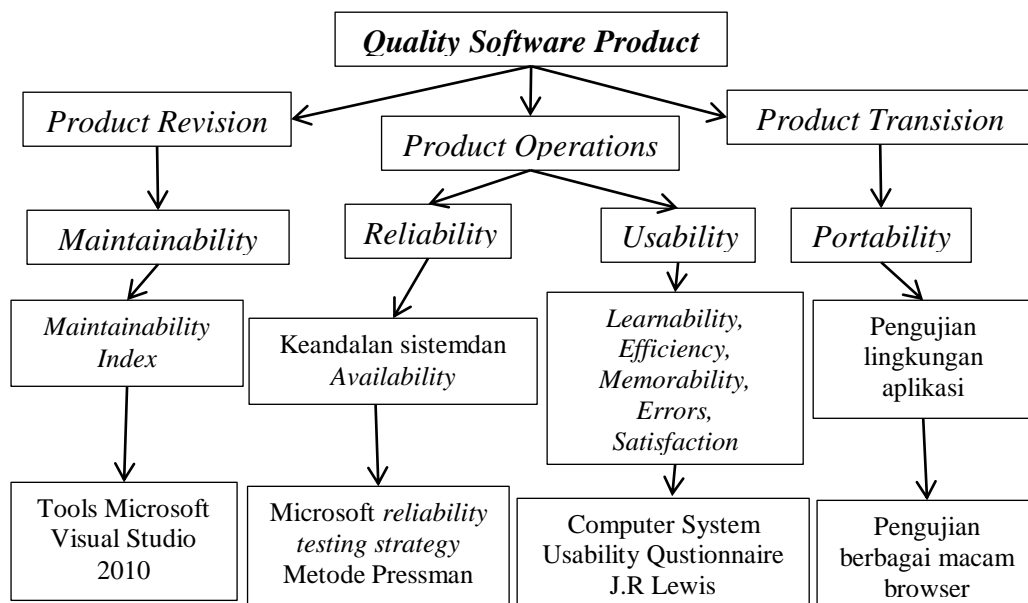
(Schach, 2008) membagi sudut pandang pencapaian *portability* perangkat lunak dalam 4 kategori, yaitu: 1) *Portable System Software*, 2) *Portable Application Software*, 3) *Portable Data*, 4) *Web-Based Applications*. Suatu kekuatan terbesar dari aplikasi berbasis web adalah bahwa aplikasi tersebut dapat mencapai tingkat portabilitas yang sangat tinggi. Pertama, dikarenakan sebuah aplikasi berbasis web dapat dibuat *portable* dengan menggunakan bahasa HTML (*Hypertext Markup Language*) atau XML (*Extensible Markup Language*) yang dapat dibaca oleh *web browser*, maka hal tersebut memungkinkan aplikasi dapat dijalankan oleh setiap pengguna. Kemudian kemampuan program aplikasi

dijalankan pada server merupakan kelebihan lain dari aplikasi berbasis *web*. Aplikasi berbasis *web* yang meletakkan programnya di server akan sangat mempermudah pengguna untuk dapat menjalankannya dengan komputer pribadinya, *smartphone*, PDA maupun *tablet*.

Portabilitas perangkat lunak berbasis *web* terhadap *web browser* yang digunakan merupakan suatu hal yang perlu diperhatikan, karena belum tentu aplikasi yang dapat berjalan di Internet Explorer dapat berjalan juga di Firefox (Schach, 2008). *Browser Statistics and Trends* (w3schools, 2013) mencatat lima *web browser* yang paling banyak digunakan selama 4 tahun lebih oleh seluruh orang di dunia tanpa digantikan posisinya oleh *browser-browser* lain. Kelima *browser* tersebut adalah Internet Explorer, Firefox, Chrome, Safari, dan Opera..

B. Kerangka Berpikir

Kerangka berpikir yang digunakan penulis digambarkan pada Gambar 2 berikut:



Gambar 2. Kerangka Berpikir

Setiap teori yang ada dalam landasan teori di atas memiliki kelebihan dan kekurangannya masing-masing. Untuk itulah penulis menggabungkan beberapa teori dari para ahli untuk disesuaikan dengan kebutuhan dari perusahaan pengembang maupun perusahaan klien. Disamping itu, karena keterbatasan yang ada pada diri penulis dan waktu yang ada, maka dalam analisis pengembangan sistem aplikasi *e-training* karyawan berbasis *web* ini hanya mengacu pada empat faktor *software quality* yakni *reliability*, *usability*, *maintainability*, dan *portability*.

Melihat bahwa menurut Ian Sommerville (Sommerville, 2003) terdapat tiga atribut kualitas kritis dalam manajemen perangkat lunak yang dapat mempengaruhi suatu sistem aplikasi, yaitu: *reliability*, *maintainability*, dan *portability*. Sedangkan dilihat dari apa yang disampaikan McCall dan rekan-rekannya, aspek *reliability* dan *usability* dianggap mewakili *major Perspectives* dari sisi *Product Operations*. Kemudian aspek *portability* dianggap telah mewakili *Product Transision*. Sedangkan aspek *maintainability* telah mewakili *Product Revision*.

Selain itu, yang juga menjadi pertimbangan adalah bahwa *reliability* dan *maintainability* dianggap telah mewakili aspek internal aplikasi. Sedangkan aspek *usability* dan *portability* dianggap mewakili aspek *eksternal* aplikasi, yakni yang berhubungan langsung dengan pengguna akhir.

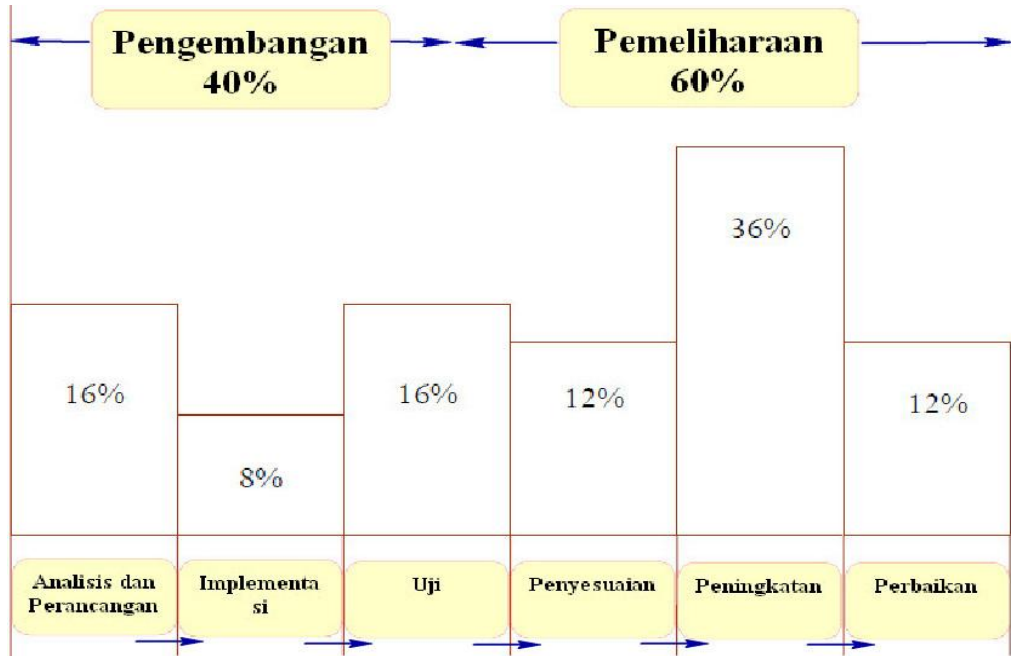
C. Hipotesis

Berdasarkan kerangka berpikir yang telah dijelaskan di atas, penulis merumuskan hipotesis penelitian. Hipotesis penelitian tersebut sebagai berikut:

1. Pengembangan sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo telah memenuhi aspek *reliability* untuk dapat digunakan oleh pengguna dengan baik.
2. Pengembangan sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo telah memiliki tingkat *usability* yang baik berdasarkan respon dari pengguna langsung.
3. Pengembangan sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo telah memenuhi aspek *maintainability* dengan baik menurut kaidah *software quality*.
4. Pengembangan sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo telah memenuhi aspek *portability* dengan baik menurut kaidah *software quality*.

BAB III METODE PENELITIAN

A. Desain Penelitian



Gambar 3. Diagram SDLC

Dari Gambar 3 Diagram *Software Development Life Cycle* (SDLC) di atas hal yang perlu ditekankan adalah bagian pengembangan (*development*) dan pemeliharaan (*maintenance*). Dua hal inilah yang nantinya akan menjadi pokok pembahasan yang akan diteliti oleh penulis. Jenis penelitian yang digunakan berdasarkan metode yang dipilih dalam analisis pengembangan sistem aplikasi *e-training* karyawan ini adalah jenis penelitian riset dan pengembangan (*research and development*).

Sugiyono dalam bukunya (Sugiyono, 2010) memberi pengertian mengenai metode penelitian dan pengembangan atau *Research and Development* (R&D) sebagai suatu metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut. Sementara jika dilihat dari

tingkat penjelasannya maka penelitian ini termasuk jenis penelitian deskriptif, karena pada rumusan masalah terdapat variabel-variabel mandiri. Masing-masing variabel tidak dibandingkan dengan sampel lain dan tidak pula dihubungkan dengan variabel lain. Desain penelitian dijabarkan menjadi empat seperti berikut:

1. Desain Penelitian Aspek *Reliability*

Penelitian untuk analisis aspek *reliability* dilakukan dengan strategi pengujian dari Microsoft sesuai yang disampaikan Lereng dan rekannya (Lereng & Fugelseth, 2004). Strategi pengujian tersebut meliputi tahapan-tahapan berikut:

a. *Component Stress Testing*

Stress testing adalah pengujian dengan cara mensimulasikan sebuah aplikasi terhadap beban kerja di luar batas (kondisi penggunaan maksimum) untuk mengetahui bagaimana aplikasi tersebut tetap dapat bekerja. *Component stress testing* hanya menguji sebagian komponen dari sistem aplikasi.

b. *Integration Stress Testing*

Setelah masing-masing komponen dilakukan uji coba (*stress testing*) hal yang selanjutnya perlu dilakukan adalah uji keseluruhan sistem. Hal ini perlu dilakukan karena biasanya antara satu komponen dengan komponen yang lain dari sistem akan saling terhubung dan berpengaruh. Uji coba ini perlu memahami apa yang mungkin dilakukan pengguna dan bagaimana pengguna melakukannya.

c. *Real World Testing*

Uji coba ini dilakukan secara penuh pada pengguna akhir selama periode 1 bulan untuk mengetahui kemungkinan kesalahan yang masih bisa terjadi. Dalam waktu 1 bulan ini juga sekaligus dilakukan perhitungan *availability*.

d. *Random Destruction Testing*

Uji coba ini dilakukan untuk mencari kesalahan-kesalahan di luar kebiasaan pengguna, seperti mengenai *input* yang tidak semestinya. Pengujian ini akan dilakukan bersama pihak pengguna dalam *User Acceptance Test (UAT)*. Sedangkan isi dari kriteria yang ada di UAT adalah kriteria-kriteria kebutuhan awal pengguna. Pengujian aspek *security* juga tidak lupa dilakukan untuk memastikan bahwa data-data yang ada dalam sistem aman. Celah yang paling sering dieksploitasi adalah *Cross Site Scripting (XSS)* dan *SQL Injection*. Celah-celah ini dapat mengakibatkan hilang atau rusaknya data, perubahan isi *website*, masuknya konten berbahaya bahkan mampu mengambil alih *host* secara keseluruhan. Pengujian *security* dilakukan dengan bantuan aplikasi Acunetix Web Vulnerability Scanner versi 8. Aplikasi ini dipilih karena aplikasi tersebut memiliki berbagai macam skenario pengujian terhadap keamanan aplikasi *web* dan juga disertai bagaimana cara memperbaiki kekurangan yang ditemukan.

2. *Desain Penelitian Aspek Usability*

Penelitian aspek *usability* dilakukan dengan cara melakukan survei terhadap pengguna akhir menggunakan angket *usability questionnaire* J.R Lewis yang dibuat dalam sebuah program aplikasi.

3. *Desain Penelitian Aspek Maintainability*

Penelitian aspek *maintainability* dilakukan dengan cara melakukan analisis terhadap baris kode sistem aplikasi *e-training* untuk mencari nilai *maintainability index*. *Tools* yang digunakan adalah Microsoft Visual Studio 2010.

4. Desain Penelitian Aspek *Portability*

Desain penelitian aspek *portability* dianalisis dengan cara menguji coba pada *web browser* yang biasa digunakan pengguna dan perangkat-perangkat yang biasa digunakan pengguna.

B. Populasi dan sampel

1. Populasi

Populasi adalah obyek atau subjek dengan karakteristik tertentu yang ditetapkan peneliti untuk dipelajari dan kemudian ditarik kesimpulan (Sugiyono, 2010). Dalam penelitian yang penulis lakukan, populasi di sini adalah semua karyawan dan beberapa manajer bagian yang menggunakan aplikasi *training*. Alasan penulis memilih populasi tersebut dikarenakan bahwa pengguna sistem aplikasi *e-training* di sini adalah para karyawan dan beberapa manajer bagian yang terkait dengan *training* di perusahaan klien. Total populasi dalam penelitian ini adalah sebanyak 500 orang karyawan termasuk manajer bagian.

2. Sampel

Sampel adalah bagian dari jumlah dan karakteristik populasi (Sugiyono, 2010). Franken menyarankan besar sampel minimum untuk penelitian jenis ini adalah sebanyak 100 orang (Franken & Wallen, 1993). Penggunaan sampel penelitian hanya digunakan pada subjek penelitian yang terkait dengan pengguna akhir sistem aplikasi *e-training*. Dalam hal ini pengguna akhir sistem aplikasi *e-training* adalah para karyawan dan beberapa manajer dari perusahaan klien. Subjek penelitian yang dijadikan sampel digunakan untuk mengukur aspek

usability. Sementara subjek penelitian lain untuk aspek *reliability*, *portability*, dan *maintainability* adalah sistem aplikasi *e-training* karyawan itu sendiri.

C. Teknik Pengumpulan Data

1. Data Primer

Untuk mendapatkan data dan informasi yang diperlukan dalam penelitian ini, maka diperlukan sebuah teknik pengumpulan data primer. Teknik pengumpulan data primer yang dilakukan antara lain:

a. Observasi

Yaitu teknik yang dilakukan dengan cara melakukan pengamatan secara langsung terhadap kegiatan perusahaan yang menyangkut objek penelitian (Guritno, Sudaryono, & Rahardja, 2011). Hal ini dilakukan untuk menunjang data-data yang telah dikumpulkan yang berhubungan dengan pembahasan. Kemudian ada pula pencatatan secara langsung dari sumber tertulis pada lokasi penelitian. Teknik observasi yang dilakukan dalam penelitian ini adalah untuk mengumpulkan data yang terkait aspek *reliability*, *portability*, dan *maintainability*. Adapun tempat observasi yang penulis pilih sebagai tempat penelitian aspek-aspek tersebut, yaitu PT. Mutiara Solusindo sebagai tim pengembang bersama penulis.

b. Wawancara

Yaitu teknik pengumpulan data dengan cara melakukan tanya jawab secara lisan terhadap bagian-bagian tertentu yang dianggap memiliki hubungan dengan materi yang sedang di susun. Adapun wawancara yang perlu dilakukan oleh penulis adalah kepada manajer bagian dari perusahaan sebagai pimpinan yang

memiliki wewenang terhadap aplikasi *e-training* sekaligus pengguna aplikasi. Kemudian ada juga karyawan-karyawan perusahaan yang berperan sebagai pengguna sistem aplikasi *e-training*.

c. Survei dengan Penyebaran Kuesioner

Yaitu teknik pengumpulan data dengan melakukan penyebaran pertanyaan berupa kuesioner untuk diisi oleh para responden. Dalam hal penyebaran kuisisioner penulis menggunakan teknik *computer delivered survey* (survei melalui media komputer) (Jogiyanto, 2008). Alasan utama menggunakan media komputer dalam melakukan survei adalah karena biaya yang murah, pertanyaan dapat dikirimkan secara lebih cepat dan hasilnya pun dapat diterima dengan lebih cepat.

Kuisisioner ditujukan untuk seluruh pengguna sistem aplikasi *e-training* dengan memperhatikan karakteristik dari para responden, karena baik manager bagian, maupun karyawan memiliki hak akses yang berbeda. Tujuan dari penyebaran kuisisioner ini adalah untuk mendapatkan data-data yang akurat dari objek penelitian tentang keadaan yang sesungguhnya terjadi terhadap sistem aplikasi *e-training* karyawan di lapangan oleh pengguna. Teknik kuisisioner ini digunakan untuk mengumpulkan data yang terkait dengan aspek *usability* aplikasi *e-training* karyawan di masing-masing pengguna sistem.

2. Data Sekunder

Dokumen - dokumen yang berhubungan dengan pembuatan sistem aplikasi *e-training* di PT. Mutiara Solusindo. Dokumen tersebut meliputi *User Requirements Document* (URD), dokumentasi program, dan dokumen *User Acceptance Testing* (UAT).

D. Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini adalah perangkat pengembangan Microsoft Visual Studio 2010, beberapa perangkat *mobile* dan personal komputer sebagai media pengembangan sekaligus pengujian terhadap aspek *reliability*, *portability*, dan *maintainability*. Instrumen penelitian lain yang digunakan dalam penelitian ini adalah angket *usability* yang mengacu pada *Computer System Usability Questionnaire* dari J.R Lewis (Lewis et al., 1995).

E. Teknik Analisis Data

Teknik analisis data untuk 4 variabel dijelaskan pada penjabaran berikut:

1. Analisis Data Aspek *Reliability*

Selain empat tahapan seperti yang ada di desain penelitian *reliability* sebelumnya. Ada satu hal lagi tambahan untuk mengetahui baik tidaknya sistem tersebut (Lereng & Fugelseth, 2004). Beberapa kriterianya adalah sebagai berikut:

- a. Telah menerapkan fitur sesuai dengan spesifikasi persyaratan dengan benar.
- b. Bebas dari kesalahan operasi dengan skenario penggunaan sistem.
- c. Memiliki tingkat keandalan yang memenuhi prasyarat awal.

Semua prasyarat keandalan sistem yang diperlukan pengguna dan juga spesifikasi kebutuhan dimasukkan pada dokumen UAT. Apabila dalam proses UAT semua kriteria dianggap lolos maka sistem aplikasi tersebut bisa dikatakan baik. Sementara analisis perhitungan mengenai *availability* yang berlangsung pada tahap *real world testing* dihitung dengan rumus berikut:

$$\text{Availability} = \text{MTTF} / (\text{MTTF} + \text{MTTR}) \times 100\%$$

Keterangan: $MTTF = \text{mean time to failure}$
 $MTTR = \text{mean time to repair}$

2. Analisis Aspek Usability

Dilakukan dengan menghitung rata-rata jawaban berdasarkan skor. Skor masing-masing pilihan dari item kuisisioner ditetapkan sebagai berikut:

- Sangat tidak setuju = 1
- Tidak setuju = 2
- Ragu-ragu = 3
- Setuju = 4
- Sangat Setuju = 5

Berdasarkan skor yang telah ditetapkan dapat dihitung sebagaimana berikut:

▪ Jumlah skor dari responden yang menjawab SS	$= \text{TotalSS} \times 5$	$=$	\downarrow Dijumlahkan
▪ Jumlah skor dari responden yang menjawab S	$= \text{TotalS} \times 4$	$=$	
▪ Jumlah skor dari responden yang menjawab RR	$= \text{TotalRR} \times 3$	$=$	
▪ Jumlah skor dari responden yang menjawab TS	$= \text{TotalTS} \times 2$	$=$	
▪ Jumlah skor dari responden yang menjawab STS	$= \text{TotalSTS} \times 1$	$=$	
Jumlah Total Skor		$=$	

Hasil jawaban dari responden sebanyak 110 orang tersebut di atas kemudian dapat dihitung nilai tertinggi dan terendah seperti berikut:

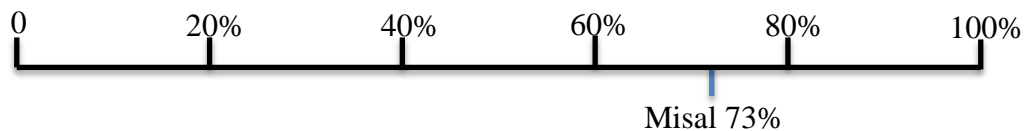
Nilai tertinggi = $\text{JumlahResponden} \times \text{jumlah item pertanyaan} \times 5 =$ (seandainya semua menjawab SS).

Nilai terendah = $\text{JumlahResponden} \times \text{jumlah item pertanyaan} \times 1 =$ (seandainya semua menjawab STS).

Nilai tertinggi yang ditemukan kemudian dijadikan acuan untuk menentukan persentase dengan rumus berikut:

Jumlah skor total / Nilai tertinggi x 100% = Misal 73%.

Dari persentase tersebut kemudian dapat dibandingkan dengan Gambar 4 berikut (Guritno et al., 2011):



Gambar 4. Tingkatan Persentase

Keterangan:

0% - 20% = Sangat Lemah

21% - 40% = Lemah

41% - 60% = Cukup

61% - 80% = Kuat

81% - 100% = Sangat Kuat

3. Analisis Aspek *Maintainability*

Faktor *maintainability* dianalisis dengan mencari nilai *maintainability index* seperti yang telah disampaikan oleh Mandeep K. Chawla (Chawla & Chhabra, 2012). Di dalam tulisannya dia menjabarkan rumus yang sama dengan yang digunakan oleh Microsoft untuk menghitung *maintainability index* di Microsoft Visual Studio 2010 (Naboulsi, 2011). Rumus perhitungannya adalah sebagai berikut:

$$MI = \text{MAX}(0, (171 - 5.2 * \ln(V) - 0.23 * (G) - 16.2 * \ln(LOC))) * 100 / 171$$

Sementara hasil dari analisis kemudian merujuk pada website resmi *Microsoft Developer Network* (MSDN, 2010). Ketentuan-ketentuannya adalah seperti yang ada dalam Tabel 4 berikut:

Tabel 4. Kategori *Maintainability*

No.	Interval	Warna	Keterangan
1	0-9	Merah	Kurang
2	10-19	Kuning	Cukup
3	20-100	Hijau	Baik

4. Analisis Aspek *Portability*

Dilakukan dengan melakukan observasi fungsionalitas komponen ketika dijalankan pada lingkungan sistem operasi dengan kedalaman *piksel* yang berbeda baik pada perangkat *mobile* maupun perangkat desktop, serta lima *web browser* yang paling banyak di gunakan di seluruh dunia.

Analisis *portability* dibatasi hanya pada *end-user portability* (lingkungan *client*). Tidak termasuk pada *system software portability* (*server*). Hal ini dikarenakan yang paling sering mengalami perubahan lingkungan adalah akses-akses yang dilakukan oleh para karyawan dan manajer (*device client*). Mereka dapat menggunakan perangkat *mobile*, *desktop* ataupun *touchscreen* yang telah disediakan perusahaan. Bukan pada sisi *server*, karena *server* tetap hanya pada satu lingkungan komputer. Suatu perangkat lunak dikatakan memiliki *portability* yang baik jika perangkat lunak tersebut dapat dijalankan pada lingkungan yang berbeda tanpa kehilangan fungsionalitasnya (Garen, 2007).

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

1. Deskripsi Data

Deskripsi data dari hasil penelitian disajikan dalam bab ini untuk memberikan gambaran secara umum mengenai penyebaran data yang diperoleh dari lapangan. Sampel yang diambil dalam penelitian ini adalah sebanyak 110 orang dari total populasi sebanyak 500 orang karyawan termasuk beberapa manajer bagian yang bersangkutan dalam hal *training*. Lokasi penelitian dilakukan di PT. Mutiara Solusindo dan perusahaan yang berperan sebagai pengguna sistem.

Penelitian dilakukan mulai tanggal 1 November 2012 sampai dengan 1 Februari 2013. Meliputi observasi sistem aplikasi *e-training* karyawan, wawancara langsung dengan pengguna maupun sesama pengembang untuk mengetahui sejauh mana penggunaan dan pengembangan sistem aplikasi *e-training* dibuat. Kemudian penyebaran kuisioner dalam bentuk program aplikasi juga dilakukan untuk mengetahui sejauh mana respon pengguna terhadap sistem yang dipakai.

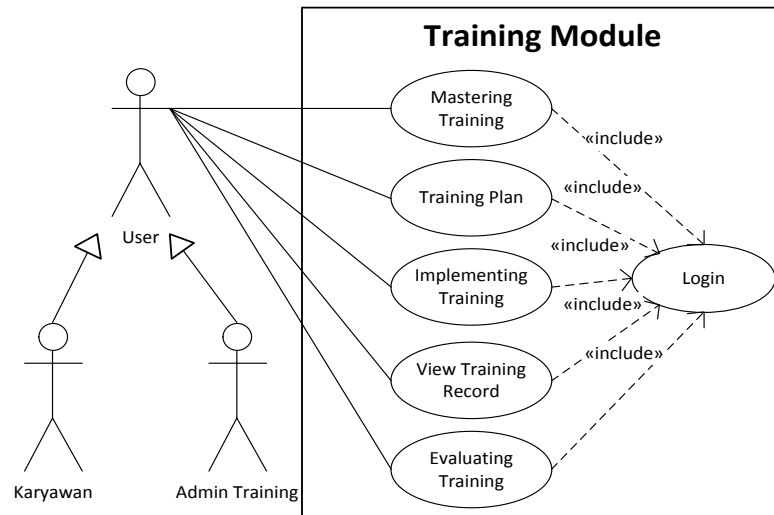
2. Deskripsi Sistem Aplikasi

Sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo yang berbasis *web* ini dibuat dengan menggunakan bahasa pemrograman ASP.NET dan C# dengan menggunakan Microsoft Visual Studio 2010. Sementara untuk

database, aplikasi ini menggunakan Microsoft SQL Server 2008. Struktur program digambarkan ke dalam pemodelan UML berikut:

a. Use Case

Fungsi-fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang menggunakannya digambarkan pada Gambar 5 berikut:



Gambar 5. Use Case Modul Training

1) Definisi Aktor

Pada Tabel 5 berikut adalah deskripsi aktor pada sistem aplikasi *e-training*.

Tabel 5. Definisi Aktor

No	Aktor	Deskripsi
1	Admin Training	Orang dalam perusahaan yang bertugas dan bertanggung jawab serta memiliki hak akses terhadap modul <i>training</i> untuk melakukan operasi pengelolaan data <i>training</i> baik mengenai waktu, tempat pelaksanaan, pembicara, partisipan sampai melakukan perekapan terhadap hasil <i>training</i> .
2	Karyawan	Orang dalam perusahaan yang berhak mengikuti <i>training</i> dan melakukan pengecekan hasil evaluasi maupun melihat semua <i>record training</i> yang telah ia ikuti selama periode tertentu. Ia juga berhak men- <i>download</i> materi yang ada dalam masing-masing topik <i>training</i> .

2) Definisi *Use Case*

Pada Tabel 6 berikut dideskripsikan masing-masing *use case* yang ada.

Tabel 6. Definisi *Use Case*

No	Use Case	Deskripsi
1	Login	Merupakan proses pengecekan hak akses, siapa yang berhak mengakses proses pengolahan data <i>training</i> dan siapa yang bisa melihat hasil proses.
2	<i>Mastering Training</i>	Merupakan proses seorang <i>admin training</i> untuk membuat sebuah acuan seperti pengaturan periode tahun, topik-topik <i>training</i> , kompetensi-kompetensi <i>training</i> , sampai standar minimal nilai yang harus diperoleh oleh peserta <i>training</i> .
3	<i>Training Plan</i>	Merupakan proses perencanaan yang dilakukan oleh seorang <i>admin training</i> untuk mewujudkan kegiatan <i>training</i> yang akan dilaksanakan. Perencanaan tersebut menyangkut tanggal, tempat, waktu pelaksanaan, partisipan yang diwajibkan hadir, dll.
4	<i>Implementing Training</i>	Merupakan proses implementasi <i>training</i> yang sebelumnya telah dilaksanakan. Seorang <i>admin training</i> akan melakukan rekap siapa saja yang hadir dalam <i>training</i> untuk kemudian menentukan apakah layak untuk dilakukan evaluasi atau tidak.
5	<i>View Training Record</i>	Merupakan proses untuk melihat semua <i>record training</i> yang telah diikuti oleh karyawan. Untuk <i>admin training</i> memiliki hak akses untuk melihat seluruh <i>record training</i> karyawannya. Sedangkan karyawan biasa hanya memiliki hak akses untuk melihat <i>record training</i> -nya sendiri.
6	<i>Evaluating Training</i>	Merupakan proses yang dilakuakn oleh <i>admin training</i> untuk melakukan evaluasi atau pemberian nilai pada masing-masing karyawannya yang telah mengikuti <i>training</i> . Kemudian menentukan apakah karyawan tersebut telah lulus atau harus mengulang suatu <i>training</i> .

3) Skenario *Use Case*

Tabel 7 berikut menjelaskan skenario yang berjalan pada *use case login*.

Tabel 7. Skenario *Use Case Login*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukan <i>username</i> , <i>password</i> , dan kode gambar	
	2. Mengecek valid tidaknya data masukan
	3. Masuk ke aplikasi pengelolaan data <i>training</i>
Skenario Alternatif	
1. Memasukan <i>username</i> , <i>password</i> , dan kode gambar	
	2. Mengecek valid tidaknya data masukan
	3. Menampilkan pesan login tidak valid
4. Memasukan <i>username</i> , <i>password</i> , dan kode gambar yang <i>valid</i>	
	5. Mengecek valid tidaknya data masukan
	6. Masuk ke aplikasi pengelolaan data <i>training</i>

Tabel 8 berikut menjelaskan sekenario yang berjalan pada *use case mastering training*.

Tabel 8. Sekenario *Use Case Mastering Training*

Aksi Aktor	Reaksi Sistem
Sekenario Normal	
1. Masukan data acuan sebagai dasar perencanaan <i>training</i>	
	2. Mengecek kelengkapan data yang dimasukan
	3. Menyimpan data yang dimasukan ke basis data
	4. Menampilkan pesan sukses disimpan
Sekenario Alternatif	
1. Masukan data acuan sebagai dasar perencanaan <i>training</i>	
	2. Mengecek kelengkapan data yang dimasukan
	3. Menampilkan pesan bahwa data yang dimasukan tidak lengkap
4. Masukan data yang kurang	
	5. Mengecek kelengkapan data yang dimasukan
	6. Menyimpan data yang dimasukan ke basis data
	7. Menampilkan pesan sukses disimpan

Tabel 9 berikut menjelaskan sekenario yang berjalan pada *use case training plan*.

Tabel 9. Skenario *Use Case Training Plan*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukan data perencanaan <i>training</i> dengan mengacu ke <i>master training</i>	
	2. Mengecek kelengkapan data yang dimasukan
	3. Menyimpan data yang dimasukan ke basis data
	4. Menampilkan pesan sukses disimpan
Skenario Alternatif	
1. Masukan data perencanaan	
	2. Mengecek kelengkapan data yang dimasukan
	3. Menampilkan pesan data kurang lengkap
4. Melengkapi kekurangan	
	5. Mengecek kelengkapan data yang dimasukan
	6. Menyimpan data yang dimasukan ke basis data
	7. Menampilkan pesan sukses disimpan

Tabel 10 berikut menjelaskan skenario yang berjalan pada *use case implementing training*.

Tabel 10. Skenario *Use Case Implementing Training*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Merubah status menjadi <i>implemented</i>	
	2. Menyimpan status baru ke basis data
3. Melakukan validasi partisipan yang hadir dan memberikan sertifikat	
	4. Menyimpan hasil implementasi ke basis data
	5. Tampilkan pesan sukses
Skenario Alternatif	
1. Merubah status menjadi <i>implemented</i>	
	2. Menyimpan status baru yang telah diubah ke basis data
3. Merubah kembali ke status <i>training plan</i>	
	4. Menyimpan perubahan ke status awal pada basis data

Tabel 11 menjelaskan skenario yang berjalan pada *use case evaluating training*.

Tabel 11. Skenario *Use Case Evaluating Training*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengevaluasi <i>training</i> yang telah diimplementasi dan menentukan apakah peserta <i>training</i> lulus atukah tidak	
	2. Menyimpan hasil evaluasi ke dalam basis data
3. Melakukan <i>closing training</i> setelah selesai dievaluasi	
	4. Memberikan tanda bahwa <i>training</i> telah dievaluasi dan menampilkan hasilnya pada masing-masing karyawan
Skenario Alternatif	
1. Mengevaluasi <i>training</i> yang telah diimplementasi dan menentukan apakah peserta <i>training</i> lulus atukah tidak	
	2. Menyimpan hasil evaluasi ke dalam basis data dan merubah status
3. <i>Roll back</i> untuk kembali ke tahap implementasi	
	4. Status kembali ke tahap implementasi
5. Mengevaluasi ulang	
	6. Menyimpan hasil evaluasi ke dalam basis data
	7. Menampilkan hasil pada masing-masing <i>record training</i> karyawan

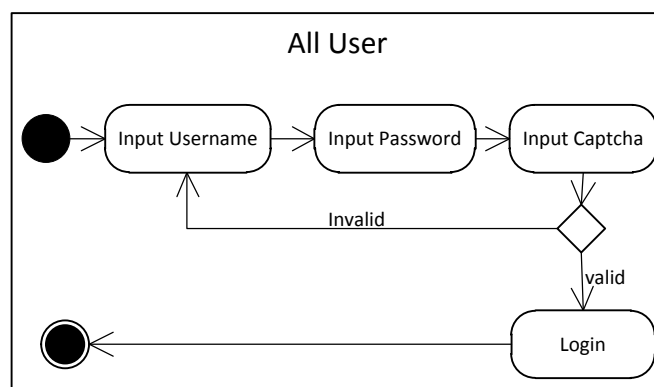
Tabel 12 berikut menjelaskan sekenario pada *use case view training record*.

Tabel 12. Sekenario *Use Case View Training Record*

Aksi Aktor	Reaksi Sistem
Sekenario Normal	
1. Lihat <i>training</i> yang telah diikuti	
	2. Tampilkan <i>record</i> pribadi (jika <i>login</i> sebagai karyawan). Tampilkan seluruh <i>record</i> karyawan (jika <i>login</i> sebagai <i>admin training</i>)
3. Mengunduh materi <i>training</i>	
	4. Melakukan <i>browse</i> menuju <i>system explorer</i> untuk menentukan lokasi penyimpanan
Sekenario Alternatif	
1. Lihat <i>training</i> yang telah diikuti	
	2. Menampilkan <i>record training</i> sesuai hak akses

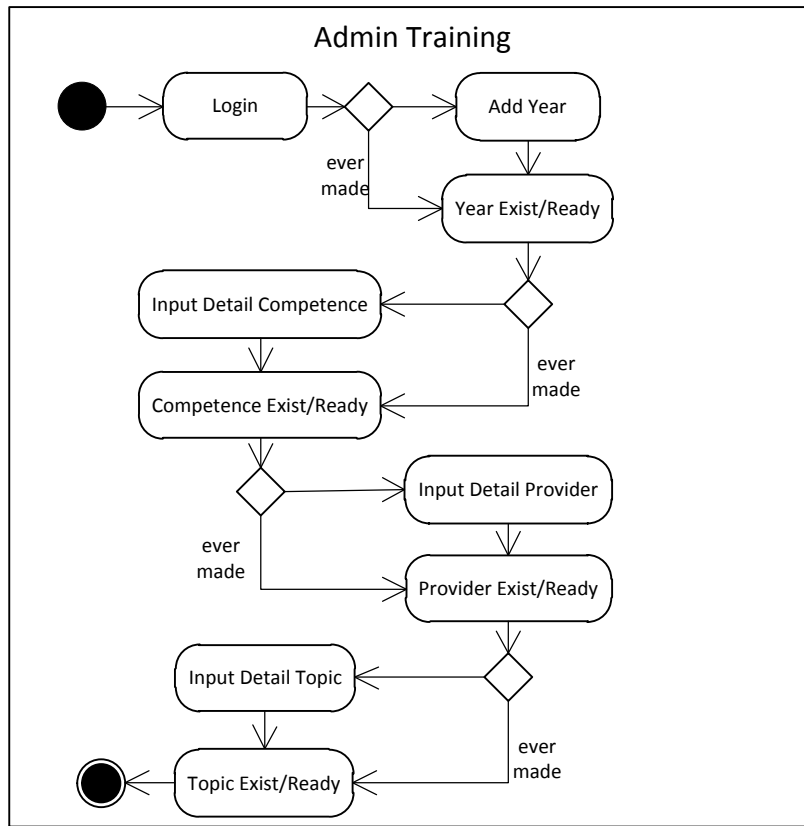
b. Activity Diagram

Aktivitas *login* digambarkan dengan *activity diagram* pada Gambar 6 berikut:



Gambar 6. *Login For All User*

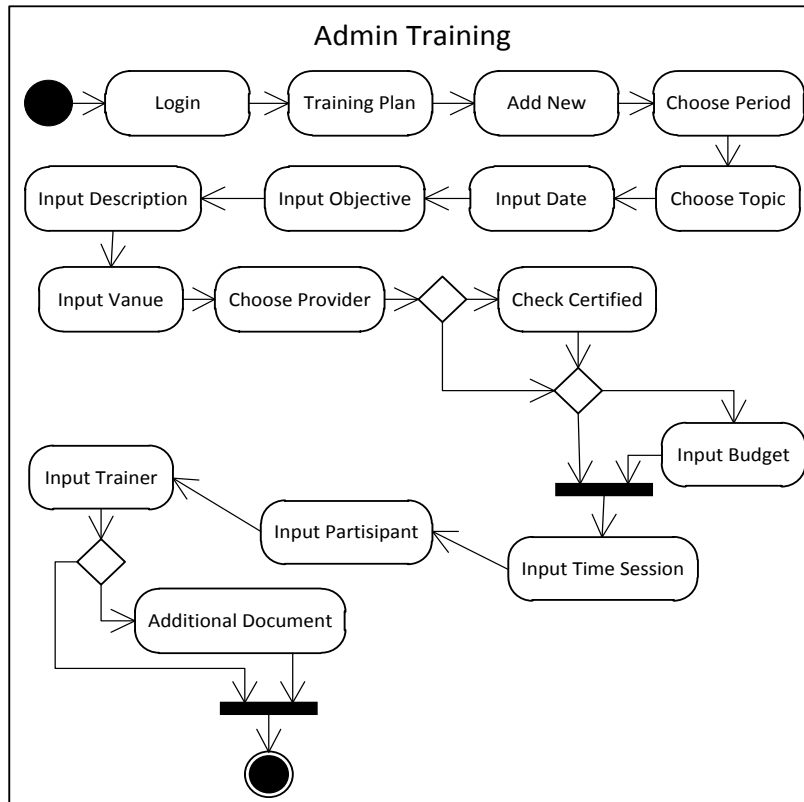
Aktivitas *preparing master training* digambarkan pada Gambar 7 berikut:



Gambar 7. Activity Diagram Preparing Master Training

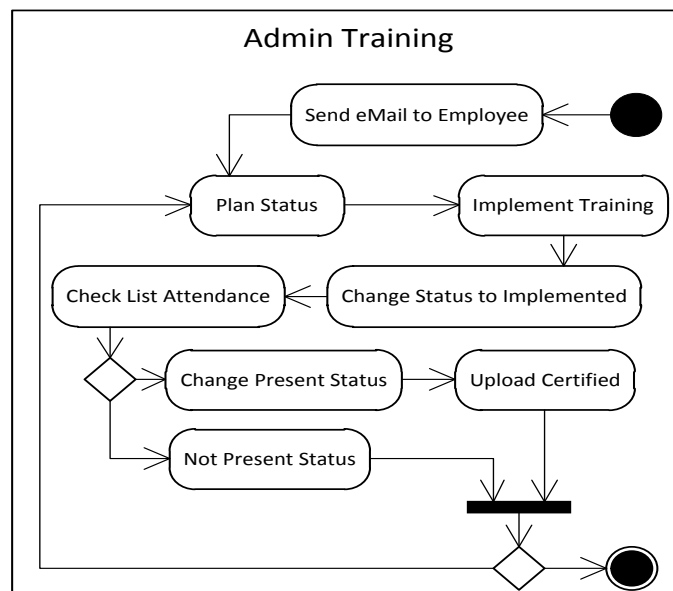
Keterangan: Asumsi bahwa seorang *admin training* telah berhasil login sehingga ia dapat melakukan penambahan tahun apabila periode tahun yang diinginkan belum ada. Apabila periode tahun yang diinginkan telah ada maka tahap setelah login adalah langsung ke “*Year Exist/Ready*”. Hampir sama dengan tahap penambahan tahun apabila *competence* yang diinginkan sudah pernah dibuat sebelumnya maka *competence* siap digunakan (*ready*) pada pembuatan *topic* baru.

Untuk aktivitas perencanaan *training* digambarkan pada Gambar 8 berikut:



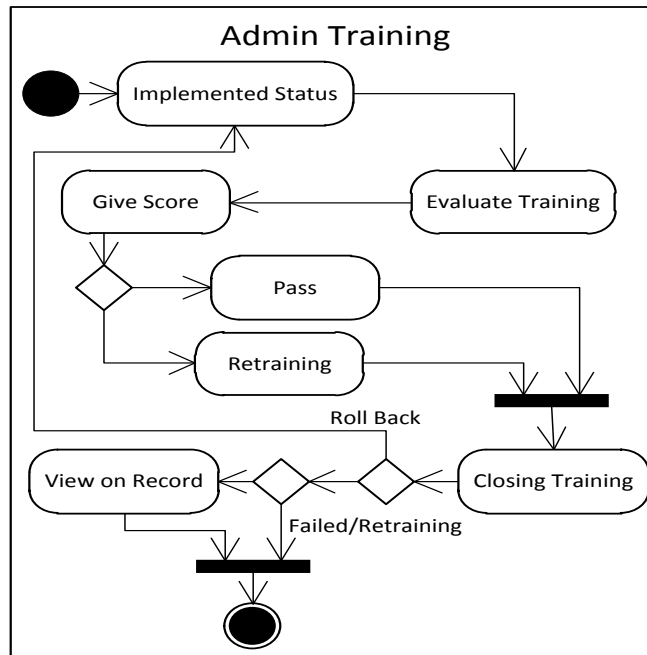
Gambar 8. Activity Diagram Training Plan

Sementara proses implementasi *training* digambarkan seperti Gambar 9 berikut:



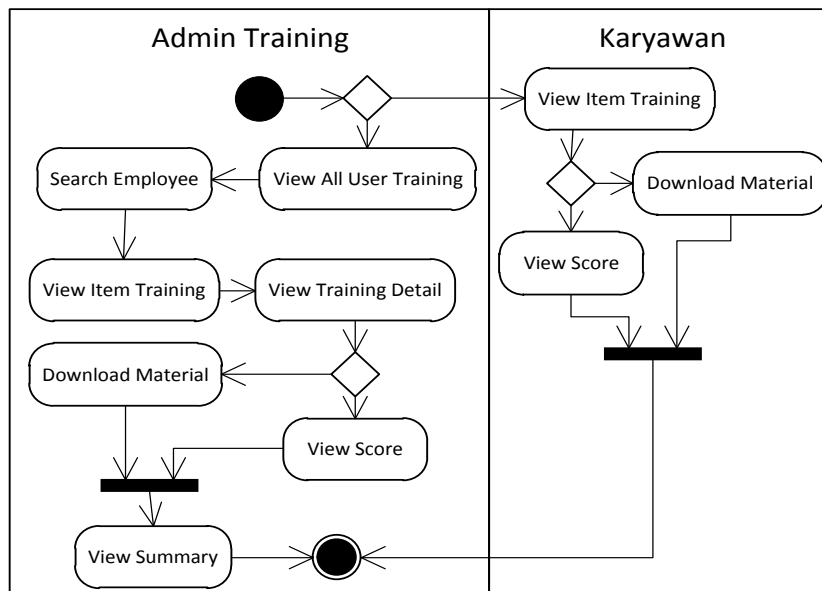
Gambar 9. Activity Diagram Implementing Training

Setelah *training* terimplementasi, selanjutnya dilakukan evaluasi *training* oleh *admin training* dengan proses seperti Gambar 10 berikut.



Gambar 10. Activity Diagram Evaluating Training

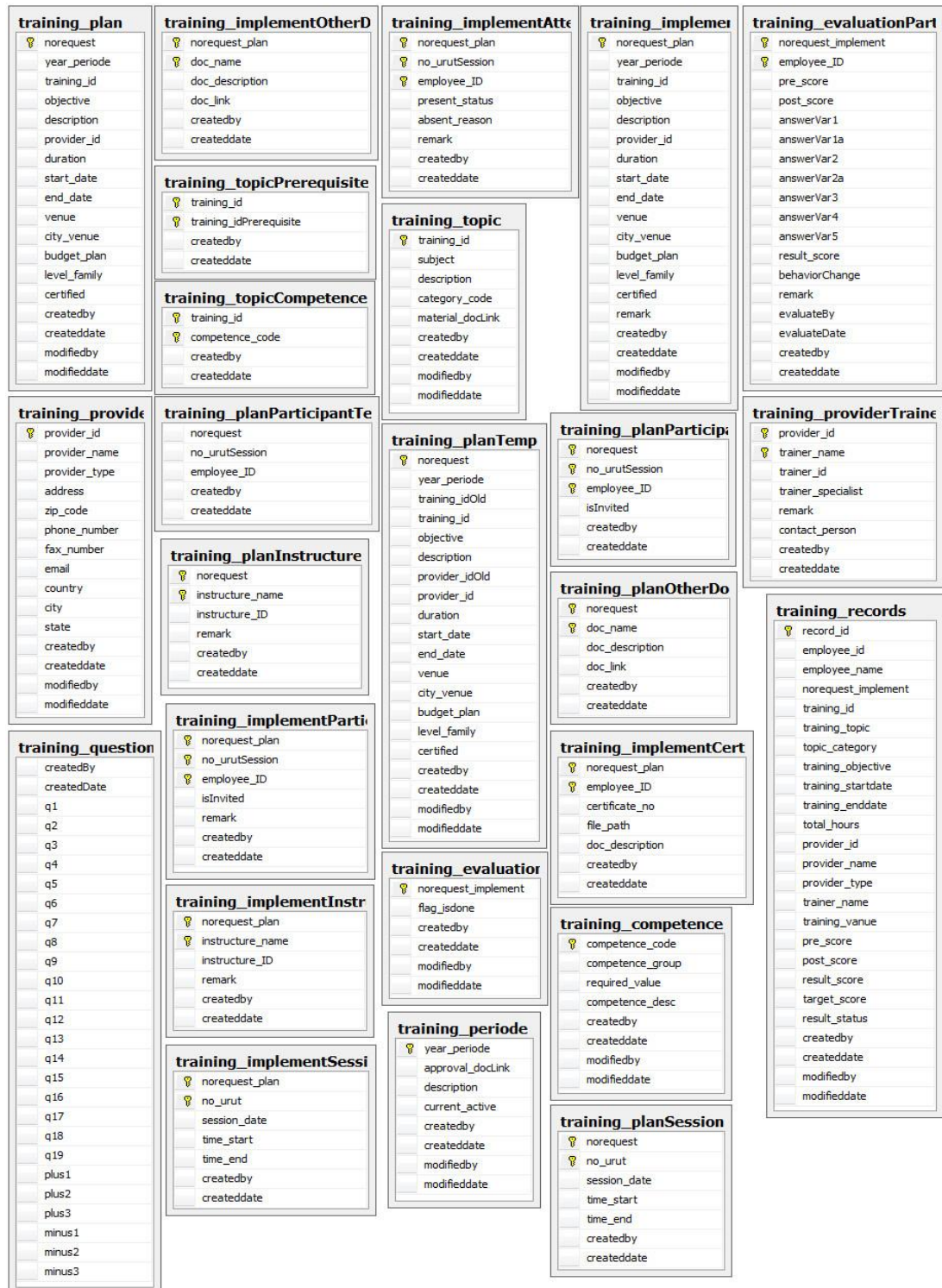
Apabila sudah dilakukan evaluasi terhadap setiap karyawan maka hasilnya dapat dilihat dengan proses seperti pada Gambar 11 berikut.



Gambar 11. Activity Diagram View Training Record

c. Tabel Database

Database aplikasi *e-training* terdiri dari tabel-tabel seperti Gambar 12 berikut.



Gambar 12. Tabel Database

B. Pembahasan

1. Reliability

Telah disebutkan bahwa menurut (Tian et al., 2003) sebuah *reliability* khususnya aplikasi berbasis web diidefinisikan sebagai “*the probability of failure...*”. *Failure* disini juga disebutkan bahwa yang dikatakan *failure* pada aplikasi berbasis *web* adalah ketidakmampuan seorang pengguna sistem untuk memperoleh atau menyampaikan informasi dengan menggunakan sistem yang dibuat, baik itu berupa dokumen maupun hasil dari komputasi sistem.

Setelah program aplikasi dianggap selesai pengujian dilakukan dengan menerapkan skenario *stress testing* 50, 70, dan 90 orang pengguna secara konstan hasilnya seperti Gambar 12 berikut:

Counter	Instance	Category	Computer	Min	Max	Avg
Key Indicators						
<input checked="" type="checkbox"/> User Load	_Total	LoadTest:Scenario	HARYSAY-PC	50	50	50
<input checked="" type="checkbox"/> Pages/Sec	_Total	LoadTest:Page	HARYSAY-PC	3.40	21.6	9.53
<input checked="" type="checkbox"/> Avg. Page Time	_Total	LoadTest:Page	HARYSAY-PC	0.0032	0.31	0.064
<input checked="" type="checkbox"/> Errors/Sec	_Total	LoadTest:Errors	HARYSAY-PC	0	1.60	0.47
<input checked="" type="checkbox"/> Threshold Violations/Sec	_Total	LoadTest:Errors	HARYSAY-PC	0	0	0
Counter	Instance	Category	Computer	Min	Max	Avg
Key Indicators						
<input checked="" type="checkbox"/> User Load	_Total	LoadTest:Scenario	HARYSAY-PC	70	70	70
<input checked="" type="checkbox"/> Pages/Sec	_Total	LoadTest:Page	HARYSAY-PC	7.80	31.8	14.9
<input checked="" type="checkbox"/> Avg. Page Time	_Total	LoadTest:Page	HARYSAY-PC	0.0036	0.41	0.080
<input checked="" type="checkbox"/> Errors/Sec	_Total	LoadTest:Errors	HARYSAY-PC	0	1.80	0.68
<input checked="" type="checkbox"/> Threshold Violations/Sec	_Total	LoadTest:Errors	HARYSAY-PC	0	0	0
Counter	Instance	Category	Computer	Min	Max	Avg
Key Indicators						
<input checked="" type="checkbox"/> User Load	_Total	LoadTest:Scenario	HARYSAY-PC	90	90	90
<input checked="" type="checkbox"/> Pages/Sec	_Total	LoadTest:Page	HARYSAY-PC	9.40	37.0	18.4
<input checked="" type="checkbox"/> Avg. Page Time	_Total	LoadTest:Page	HARYSAY-PC	0.0039	0.56	0.100
<input checked="" type="checkbox"/> Errors/Sec	_Total	LoadTest:Errors	HARYSAY-PC	0	2.00	1.05
<input checked="" type="checkbox"/> Threshold Violations/Sec	_Total	LoadTest:Errors	HARYSAY-PC	0	0	0

Gambar 13. Hasil Uji *Stress Testing*

Keterangan:

- User load = Jumlah pengguna dalam skenario pengujian.
- Pages/sec = Beban halaman per detik yang diakses oleh *user load*.
- Avg. Test Time = Waktu rata-rata untuk melakukan download halaman dan berbagai permintaan seperti gambar, css, js, dll
- Errors/sec = Kesalahan yang terjadi per detik
- Threshold Violation/Sec = Ambang batas sistem tetap dapat berjalan dalam beban uji yang dilakukan

Hasil tes di atas menunjukkan meski dalam kondisi beban kerja yang tinggi sistem aplikasi tidak menunjukkan adanya *critical threshold* sehingga tidak ada yang perlu dikhawatirkan. Kemudian tes yang dilakukan ketika beban kerja tersebut berlangsung juga menunjukkan bahwa sistem tetap dapat bekerja dengan baik termasuk akses *database* yang lancar.

Setelah dalam pengujian tersebut di atas dianggap lolos maka kemudian sistem aplikasi diuji coba pada pengguna secara langsung dalam periode 1 bulan. Hasil dari uji coba selama 1 bulan tersebut dapat dilihat pada Tabel 13 berikut:

Tabel 13. Daftar Kesalahan dan Perbaikan

No	Kesalahan	Perbaikan (Hari)	TTF
1	Keterangan lokasi ruangan <i>training</i> tidak dapat muncul (04/01/2013)	1	2
		7/1/2012	
2	Gagal melakukan upload materi /Manual <i>Training</i> (Softfile Excel, Powerpoint, PDF, dll), Sertifikat <i>Training</i> pada Individual <i>Training Record</i> tidak bisa diupload(17/01/2013)	2	8
		18-21/01/12	
3	Notifikasi email pada H-7, H-3 dan H-1 sebelum <i>training</i> tidak dapat berjalan, (30/01/2013)	1	5
		31/01/12	
Rata-rata		1,3	5

Sebagai tambahan analisis *reliability* seperti yang disampaikan Pressman maka dilakukanlah perhitungan *availability*. Nilai *availability* didapat dari hasil perhitungan MTTF (*mean time to failure*) dan MTTR (*mean time to repair*) pada pengembangan sistem aplikasi *e-training* karyawan. Berdasarkan kesalahan yang terjadi seperti tersebut maka diperoleh perhitungan sebagai berikut:

Setiap hari, sistem bekerja dari jam 08.00-17.00, ± 9 jam/hr. Jumlah jam selama sistem dijalankan dalam 1 bulan (tgl 2 jan – 1 feb) = 13hr. 13 x 9 jam = 117 jam.

$$Availability = \frac{MTTF}{(MTTF+MTTR)} \times 100$$

$$Availability = 5 / (5 + 1,3) \times 100$$

$$= 5 / 6,33 \times 100$$

$$= 0,79 \times 100$$

$$= 79 \%$$

Kriteria baik dan tidaknya nilai *availability* berdasarkan presentase hasil perhitungan mengacu pada skala yang dikemukakan oleh Suryo Guritno (Guritno et al., 2011). Pembagian skala ini memperhatikan rentang bilangan dengan kondisi maksimal yang diharapkan adalah 100%, sehingga persentase antara 0-100 dibagi rata menjadi lima kriteria seperti Tabel 14 berikut:

Tabel 14. Kategori Nilai *Availability*

No.	Skor dalam Persentase	Kategori
1	81% - 100%	Baik Sekali
2	61% - 80%	Baik
3	41% - 60%	Cukup
4	21% - 40%	Kurang
5	0% - 20%	Sangat Kurang

Hasil perhitungan *availability* berada pada kategori nomor 2, hal ini memiliki makna bahwa dari pemakaian selama 1 bulan, 79% fungsionalitas atau ketersediaan sistem masih dapat berjalan dengan baik.

Kemudian hasil dari pengujian *User Acceptance Test* (UAT) yang terdapat di lampiran menunjukkan bahwa setelah dilakukannya perbaikan terakhir, tidak ditemukan lagi kesalahan. Dari sini dapat disimpulkan bahwa semua fungsi yang dibutuhkan dapat berjalan dengan baik tanpa terjadi *error* dan sistem dapat memenuhi kebutuhan pengguna. (Tian et al., 2003). Sementara dari segi *security* yang diuji diperoleh hasil bahwa celah keamanan yang ditemukan memiliki tingkat ancaman yang tidak begitu dikhawatirkan seperti Tabel 15 berikut:

Tabel 15. Hasil Pengujian Security

No.	Jenis Celah Keamanan	Tingkat Peringatan
1	<i>User credentials are sent in clear text</i>	<i>Low</i>
2	<i>Password type input with autocomplete enabled</i>	<i>Informational</i>

2. Usability

Setelah melakukan survei dengan menyebarkan kuisisioner berupa program aplikasi yang menjadi satu dengan *e-training* karyawan, maka diperoleh data sebanyak 110 responden. Tabel 16 berikut adalah hasil dari jawaban para responden mengenai kuisisioner yang diberikan.

Tabel 16. Hasil Jawaban Kuisisioner

Pertanyaan	Jawaban Responden				
	Sangat Setuju (SS)	Setuju (S)	Ragu-Ragu (RR)	Tidak Setuju (TS)	Sangat Tidak Setuju (STS)
P1	15	60	35	0	0
P2	34	57	14	5	0
P3	18	61	31	0	0
P4	8	55	47	0	0
P5	30	68	12	0	0
P6	44	61	5	0	0
P7	33	61	14	2	0
P8	14	53	40	3	0
P9	27	58	21	4	0
P10	34	54	17	5	0
P11	5	58	45	2	0
P12	52	57	1	0	0
P13	30	65	15	0	0
P14	35	60	12	3	0
P15	41	66	3	0	0
P16	39	57	14	0	0
P17	53	57	0	0	0
P18	8	59	41	2	0
P19	58	44	8	0	0
Jumlah	578	1111	375	26	0

Dari data yang diperoleh tersebut kemudian dianalisis dengan menghitung rata-rata jawaban berdasarkan skor yang diperoleh dari setiap jawaban responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagaimana berikut:

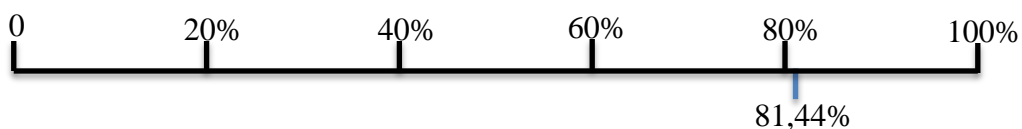
▪ Jumlah skor dari responden yang menjawab SS	= 578 x 5	= 2890
▪ Jumlah skor dari responden yang menjawab S	= 1111 x 4	= 4444
▪ Jumlah skor dari responden yang menjawab RR	= 375 x 3	= 1125
▪ Jumlah skor dari responden yang menjawab TS	= 26 x 2	= 52
▪ Jumlah skor dari responden yang menjawab STS	= 0 x 1	= 0
<hr/>		
Jumlah Total		= 8511

Hasil jawaban dari responden sebanyak 110 orang tersebut di atas kemudian dapat dihitung nilai tertinggi dan terendah seperti berikut:

Nilai tertinggi = $110 \times 19 \times 5 = 10450$ (seandainya semua menjawab SS).

Nilai terendah = $110 \times 19 \times 1 = 2090$ (seandainya semua menjawab STS).

Berdasarkan perhitungan yang menyatakan nilai tertinggi adalah 10450 dapat dicari persentase seperti berikut: $8511 / 10450 \times 100\% = 81,44\%$. Dari persentase tersebut kemudian dapat diketahui bahwa tingkat *usability* sistem aplikasi *e-training* karyawan berdasarkan persepsi pengguna tergolong sangat kuat. Hal ini dapat dilihat dari Gambar 14 berikut:



Gambar 14. Tingkatan Persentase

Keterangan:

0% - 20% = Sangat Lemah

61% - 80% = Kuat

21% - 40% = Lemah

81% - 100% = Sangat Kuat

41% - 60% = Cukup

Identifikasi kecenderungan skor data angket ditetapkan berdasarkan kriteria ideal. Berdasarkan data penskoran skala likert untuk 19 butir pertanyaan diperoleh skor ideal terendah (1×19) = 19 dan skor ideal tertinggi (5×19) = 95. Rerata ideal (M_i) dan simpangan baku ideal (S_{bi}) dapat dihitung dengan acuan norma sebagai berikut:

$$M_i = \frac{1}{2} (\text{Skor tertinggi} + \text{skor terendah})$$

$$= \frac{1}{2} (95+19) = 57$$

$$S_{bi} = \frac{1}{6} (\text{Skor tertinggi} - \text{skor terendah})$$

$$= \frac{1}{6} (95-19)$$

$$= \frac{1}{6} \times 76$$

$$= 12,7$$

Kecenderungan dari hasil angket *usability* dapat dilihat pada skor rerata ideal (X) dengan ketentuan sebagai berikut:

$$M_i + 1.S_{bi} = 57+12,7 = 69,7$$

$$M_i - 1.S_{bi} = 57-12,7 = 44,3$$

Tabel 17. Kategorisasi Kecenderungan Skor

No.	Kategori	Interval	Interpretasi	f	f (%)
1	$X \geq M_i + 1.S_{bi}$	$X \geq 69,7$	Sangat Puas	99	90
2	$M_i + 1.S_{bi} > X \geq M_i$	$69,7 > X \geq 57$	Puas	11	10
3	$M_i > X \geq M_i - 1.S_{bi}$	$57 > X \geq 44,3$	Cukup Puas	0	0
4	$X < M_i - 1.S_{bi}$	$X < 44,3$	Kurang Puas	0	0
	Jumlah			110	100

Berdasarkan kategori skor di Tabel 17 dapat diketahui bahwa kepuasan terhadap sistem aplikasi *e-training* karyawan 99 subjek (90%) merasa sangat puas. Skor tersebut jika diinterpretasikan berada pada kategori yang sangat baik.

3. Maintainability

Pengukuran *maintainability* dilakukan dengan menggunakan *tools* dari Microsoft

Visual Studio 2010, hasilnya seperti pada Gambar 15 berikut untuk *layer* model:

Hierarchy	Maintainability Index	Cyclomatic Complexity	Lines of Code	Depth of Inheritance	Class Coupling
UserManagement	91	18	18	1	2
Training_topicPrerequisite	90	14	14	1	3
Training_topicCompetence	90	14	14	1	3
Training_topic	89	29	29	1	3
Training_records	88	74	74	1	3
Training_questionnaire	88	83	83	1	3
Training_providerTrainer	89	26	26	1	3
Training_provider	88	47	47	1	3
Training_planSession	89	23	23	1	3
Training_planParticipant	89	20	20	1	3
Training_planOtherDoc	89	20	20	1	3
Training_planInstructure	89	20	20	1	3
Training_plan	88	56	56	1	4
Training_periode	89	23	23	1	3
Training_implementSession	89	23	23	1	3
Training_implementParticipant	89	23	23	1	3
Training_implementOtherDoc	89	20	20	1	3
Training_implementInstructure	89	20	20	1	3
Training_implementCertificate	89	23	23	1	3
Training_implementAttendance	89	26	26	1	3
Training_implement	88	59	59	1	4
Training_evaluationParticipant	88	56	56	1	3
Training_evaluation	89	20	20	1	3
Training_competence	89	26	26	1	3
Training_processHistory	87	389	389	1	4

Gambar 15. MI Model

Sedangkan untuk *data layer* ditampilkan pada Gambar 16 berikut:

Hierarchy	Maintainability Index	Cyclomatic Complexity	Lines of Code	Depth of Inheritance	Class Coupling
UserManagementDao	70	14	72	1	6
Training_topicPrerequisiteDao	70	9	43	1	7
Training_topicDao	67	9	52	1	7
Training_topicCompetenceDao	70	9	43	1	7
Training_reportDao	68	8	52	1	5
Training_recordsDao	63	12	97	1	7
Training_questionnaireDao	60	4	41	1	6
Training_providerTrainerDao	68	9	51	1	7
Training_providerDao	64	9	64	1	7
Training_planSessionDao	70	11	49	1	8
Training_planParticipantDao	70	9	45	1	7
Training_planOtherDocDao	70	9	42	1	7
Training_planInstructureDao	70	10	52	1	7
Training_planDao	65	15	97	1	8
Training_periodeDao	68	8	42	1	7
Training_implementSessionDao	68	12	60	1	8
Training_implementParticipantDao	68	10	56	1	7
Training_implementOtherDocDao	70	8	38	1	7
Training_implementInstructureDao	70	10	52	1	7
Training_implementDao	63	9	70	1	8
Training_implementCertificateDao	68	10	54	1	7
Training_implementAttendanceDao	68	12	65	1	7
Training_evaluationParticipantDao	64	10	74	1	7
Training_evaluationDao	70	9	47	1	7
Training_competenceDao	68	8	44	1	7
Training_processHistoryDao	44	11	305	1	7

Gambar 16. MI Data Layer

Kemudian layer selanjutnya yaitu *business layer* diperoleh perhitungan *maintainability index* seperti pada Gambar 17 berikut:

Hierarchy	Maintainability Index	Cyclomatic Complexity	Lines of Code	Depth of Inheritan...	Class Coupling
▶ UserManagementBL	78	12	36	1	3
▶ Training_topicPrerequisiteBL	78	6	18	1	3
▶ Training_topicCompetenceBL	78	6	18	1	3
▶ Training_topicBL	78	6	18	1	3
▶ Training_reportBL	77	6	18	1	3
▶ Training_recordsBL	78	9	27	1	4
▶ Training_questionnaireBL	79	2	6	1	3
▶ Training_providerTrainerBL	78	6	18	1	3
▶ Training_providerBL	78	6	18	1	3
▶ Training_planSessionBL	78	8	24	1	3
▶ Training_planParticipantBL	78	6	18	1	3
▶ Training_planOtherDocBL	78	6	18	1	3
▶ Training_planInstructureBL	78	7	21	1	3
▶ Training_planBL	76	14	43	1	13
▶ Training_periodeBL	78	6	18	1	3
▶ Training_implementSessionBL	78	8	24	1	3
▶ Training_implementParticipantBL	78	7	21	1	3
▶ Training_implementOtherDocBL	78	5	15	1	3
▶ Training_implementInstructureBL	78	7	21	1	3
▶ Training_implementCertificateBL	78	7	21	1	3
▶ Training_implementBL	78	6	18	1	3
▶ Training_implementAttendanceBL	78	8	24	1	3
▶ Training_evaluationParticipantBL	78	7	21	1	3
▶ Training_evaluationBL	78	6	18	1	3
▶ Training_competenceBL	78	5	15	1	3
▶ Tra_navroll_process_historyBL	77	7	21	1	3

Gambar 17. MI Business Layer

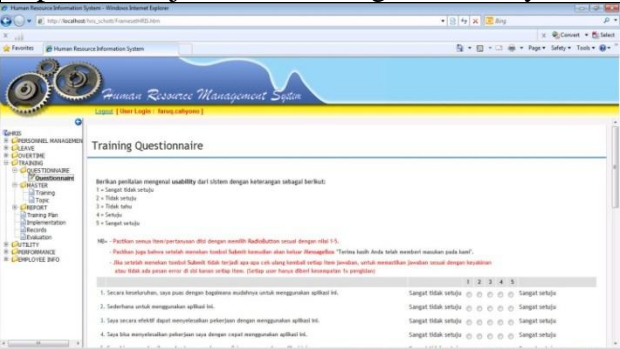

Menurut (MSDN, 2010) melalui website resminya menyatakan bahwa apabila *maintainability index* berwarna hijau atau nilainya antara 20-100 maka program tersebut dikatakan memiliki *maintainability* yang baik. Apabila *maintainability index* bernilai antara 10-19 atau berwarna kuning, maka berarti bahwa program tersebut memiliki *maintainability* yang cukup. Sementara apabila *maintainability index* bernilai antara 0-9 atau berwarna merah hal tersebut berarti bahwa program tersebut memiliki *maintainability* yang rendah.

Karena hasil analisis *maintainability index* dari sistem aplikasi *e-training* memiliki nilai antara 20-100 maka dapat dikatakan bahwa sistem telah memenuhi aspek *maintainability* menurut kaidah *software quality* dengan baik.

4. Portability

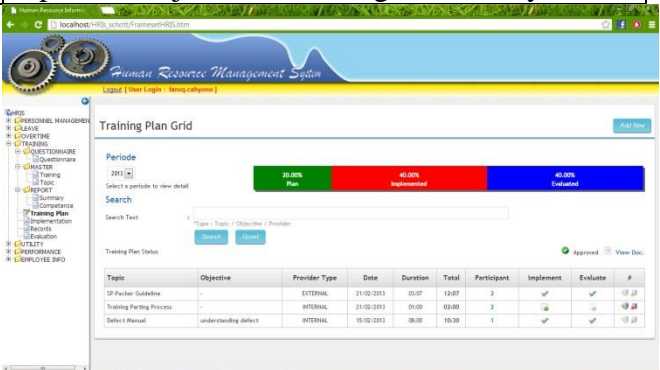

Aspek *portability* dianalisis dengan melakukan pengujian terhadap program aplikasi pada berbagai *browser* yang berbeda seperti pada Tabel 18 berikut:

Tabel 18. Pengujian *Portability Browser* & Resolusi 1

No	Pengujian Faktor Kualitas <i>Portability</i> (<i>Browser</i>)		Hasil
1	Nama tes	PortabilityTest1	Sukses
	Tujuan	Menguji aplikasi berjalan di Internet Explorer	
	Asumsi	Aplikasi telah terinstal di server dan perangkat pengguna dapat mengakses server	
	Resolusi /Browser	PC 1366 x 768 /Internet Explorer	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi berjalan sesuai fungsionalitasnya	
	Hasil pengamatan	Aplikasi berjalan sesuai fungsionalitasnya	
	Screenshoot		
2	Nama tes	PortabilityTest2	Sukses
	Tujuan	Menguji aplikasi berjalan di Firefox	
	Asumsi	Aplikasi telah terinstal di server dan perangkat pengguna dapat mengakses server	
	Resolusi /Browser	PC 1366 x 768 /Firefox	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi berjalan sesuai fungsionalitasnya	
	Hasil pengamatan	Aplikasi berjalan sesuai fungsionalitasnya	
	Screenshoot		

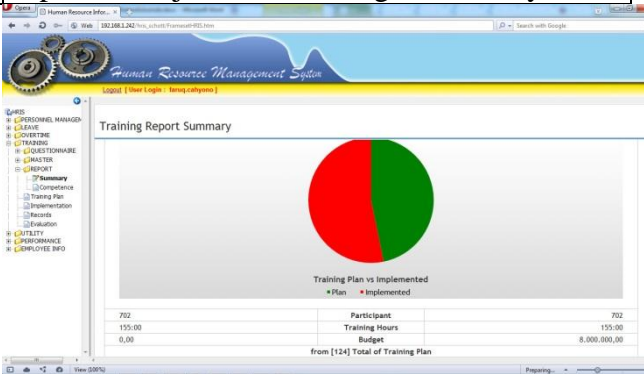

Browser dengan sistem operasi yang berbedapun ikut dilakukan pengujian seperti yang terlihat pada Tabel 19 berikut:

Tabel 19. Pengujian *Portability Browser & Resolusi 2*

No	Pengujian Faktor Kualitas <i>Portability (Browser)</i>		Hasil
3	Nama tes	PortabilityTest2	Sukses
	Tujuan	Menguji aplikasi berjalan di Chrome	
	Asumsi	Aplikasi telah terinstal di server dan perangkat user dapat mengakses server	
	Resolusi /Browser	PC 1366 x 768 /Chrome	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi berjalan sesuai fungsionalitasnya	
	Hasil pengamatan	Aplikasi berjalan sesuai fungsionalitasnya	
	Screenshot		
4	Nama tes	PortabilityTest3	Sukses
	Tujuan	Menguji aplikasi berjalan di Safari	
	Asumsi	Aplikasi telah terinstal di server dan perangkat user dapat mengakses server	
	Resolusi/Browser	PC 1024 x 768 / Safari	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi berjalan sesuai fungsionalitasnya	
	Hasil pengamatan	Aplikasi berjalan sesuai fungsionalitasnya	
	Screenshot		

Kemudian dilakukan pengujian juga terhadap browser perangkat mobile smartphone seperti seperti yang terlihat pada Tabel 20 berikut:

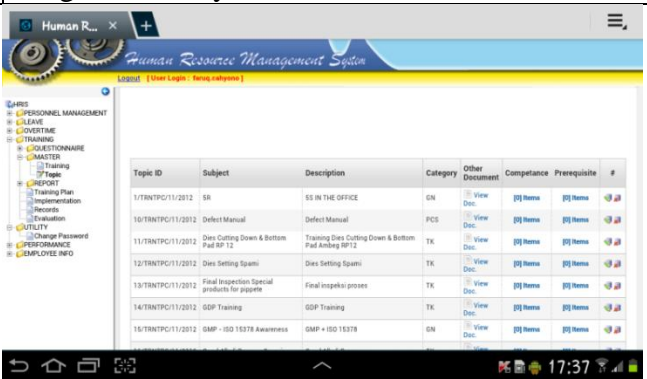
Tabel 20. Pengujian *Portability Browser & Device*

No	Pengujian Faktor Kualitas <i>Portability (Device)</i>		Hasil
5	Nama tes	PortabilityTest5	Sukses
	Tujuan	Menguji aplikasi berjalan di Opera	
	Asumsi	Aplikasi telah terinstal di server dan perangkat user dapat mengakses server	
	Resolusi/ Browser	PC 1366 x 768 / Opera	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi berjalan sesuai fungsionalitasnya	
	Hasil pengamatan	Aplikasi berjalan sesuai fungsionalitasnya	
	Screenshoot		
6	Nama tes	PortabilityTest6	Sukses
	Tujuan	Menguji aplikasi pada browser smartphone	
	Asumsi	Aplikasi telah terinstal di server dan perangkat user dapat mengakses server	
	Device	Smartphone	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi berjalan sesuai fungsionalitasnya	
	Hasil pengamatan	Aplikasi berjalan sesuai fungsionalitasnya	
	Screenshoot		

Terakhir pengujian dilakukan pada perangkat tablet dengan hasil seperti pada

Tabel 21 berikut:

Tabel 21. Pengujian *Portability Device* & Resolusi

No	Pengujian Faktor Kualitas <i>Portability (Device)</i>		Hasil
7	Nama tes	PortabilityTest7	Sukses
	Tujuan	Menguji apakah aplikasi dapat berjalan pada perangkat mobile tablet	
	Asumsi	Aplikasi telah terinstal di server dan perangkat user dapat mengakses server	
	Device/Resolusi	Tablet / 600 x 1024	
	Langkah	Panggil aplikasi melalui browser	
	Hasil diharapkan	Aplikasi dapat berjalan sebagaimana mestinya tanpa kehilangan fungsionalitasnya	
	Hasil pengamatan	Aplikasi dapat berjalan tanpa kehilangan fungsionalitasnya	
	Screenshot		

Karena program aplikasi *e-training* dapat berjalan di setiap lingkungan yang berbeda (5 browser terkenal dan device berbeda) tanpa kehilangan fungsionalitasnya dan dapat memenuhi kriteria-kriteria yang didefinisikan oleh (Schach, 2008) mengenai *portability* aplikasi berbasis web, maka dapat disimpulkan bahwa sistem aplikasi *e-training* yang ada di PT. Mutiara Solusindo memiliki *portability* yang baik.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa pengembangan sistem aplikasi *e-training* karyawan berbasis *web* pada PT. Mutiara Solusindo telah memenuhi masing-masing aspek dengan baik menurut kaidah *software quality*. Hasil yang diperoleh dari masing-masing aspek yang diteliti adalah sebagai berikut:

1. Hasil analisis pengembangan sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo dilihat dari aspek *reliability* memiliki kategori yang baik karena hasil uji coba kesalahan pada *stress testing* menunjukkan nilai 0 dan hasil UAT juga menunjukkan kesesuaian terhadap kebutuhan dari pengguna. Kemudian didukung juga dengan nilai *availability* yang didapat sebesar 79% ketika sebelum dilakukan UAT.
2. Hasil analisis pengembangan sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo dilihat dari aspek *usability* memiliki tingkat respon baik yang sangat kuat dengan hasil 81,44% dan juga dengan kecenderungan skor yang menyatakan sangat puas sebesar 90%.
3. Hasil analisis pengembangan sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo dilihat dari aspek *maintainability* memiliki kategori yang baik karena semua kode program memiliki nilai *maintainability index* (MI) di atas 20.

4. Hasil analisis pengembangan sistem aplikasi *e-training* karyawan pada PT. Mutiara Solusindo dilihat dari aspek *portability* memiliki kategori yang baik karena dapat berjalan pada minimal 5 *web browser* yang biasa dipakai pengguna baik di lingkungan perusahaan klien maupun di seluruh dunia. Sistem aplikasi juga dapat berjalan dengan baik tanpa kehilangan fungsionalitasnya pada beberapa perangkat yang berbeda.

B. Saran

Mengingat berbagai keterbatasan yang dimiliki penulis baik dari sisi pemikiran maupun waktu, maka penulis menyarankan untuk analisis pengembangan sistem selanjutnya dapat mencakup aspek-aspek yang lain seperti *security*, *actuality*, *efficiency* maupun yang terlepas dari analisis aspek kualitas seperti analisis desain aplikasi. Kemudian mengenai sistem yang diteliti, alangkah baiknya jika tidak hanya sistem aplikasi *e-training* saja yang dianalisis, melainkan semua aplikasi yang ada di PT.Mutiara Solusindo.

DAFTAR PUSTAKA

- Chandra, M. (2012). 77 % Karyawan Indonesia Mendapat Training Tahun Ini. Retrieved 5 November, 2012
- Chawla, M. K., & Chhabra, I. (2012). Implementing Source Code Metrics for Software quality analysis. *International Journal of Engineering Research & Technology*, 1(5).
- Coleman, D., Ash, D., Lowther, B., & Oman, P. (1994). Using Metrics to Evaluate Software System Maintainability. 27, 44-49.
- Franken, J. R., & Wallen, N. E. (1993). *How To Design and Evaluate in Education*. USA: McGraw-Hill.
- Garen, K. (2007). Software Portability: Weighing Options, Making Choices. *The CPA Journal*, 77(11).
- Guritno, S., Sudaryono, & Rahardja, U. (2011). *Theory and Application of IT Research: Metodologi Penelitian Teknologi Informasi*. Yogyakarta: Andi.
- ISO9126. (2004). Software engineering – Product quality; Parts 1–4.
- Jogiyanto. (2008). *Metodologi Penelitian Sistem Informasi*. Yogyakarta: Andi.
- Kappel, G., Proll, B., Reich, S., & Retschitzegger, W. (2006). *Web Engineering: The Discipline of Systematic Development of Web Applications*. Hoboken, New Jersey: John Wiley & Sons Ltd.
- Ladjamudin, A.-B. B. (2006). *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu.
- Lereng, S. F., & Fugelseth, L. (2004). *Testing of Web-based Systems*. Norwegia: NTNU.
- Lewis, J. R., Group, H. F., & Boca Raton, F. (1995). IBM Computer Usability Satisfaction Questionnaires:
- McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factor in Software Quality: Concept and Definitions of Software Quality*. New York: Griffiss Air Force Base.
- MSDN. (2010). Code Metrics Values. Retrieved 26 December, 2012
- Naboulsi, Z. (2011). Code Metrics – Maintainability Index. Retrieved 27 December, 2012

- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann, USA: Morgan Kaufmann.
- Oman, P., & Hagemester, J. (1992, 9-12 Nov 1992). *Metrics for assessing a software system's maintainability*. Paper presented at the Software Maintenance, 1992. Proceedings., Conference on.
- Pressman, R. S. (2012). *Rekayasa Perangkat Lunak: pendekatan praktisi (Buku Satu)* (G. J. L. N. Adi Nugraha, Theresia Herlina Rochadiani, Ike Kurniawati, Trans.). Yogyakarta: Andi.
- Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction*, 7(1), 57-78. doi: 10.1080/10447319509526110
- S., R. A., & Shalahudin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula.
- Schach, S. R. (2008). *Object-Oriented Software Engineering*. New York, USA: McGraw-Hill.
- Sommerville, I. (2003). *Software Engineering (Rekayasa Perangkat Lunak)* (Y. Hanum, Trans.). Jakarta: Erlangga.
- Sugiyono. (2010). *Metode Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif, dan R&D)*. Bandung: Alfabeta.
- Tian, J., Ma, L., Li, Z., & Koru, A. G. u. (2003). A Hierarchical Strategy for Testing Web-Based Applications and Ensuring their Reliability. *COMPSAC 27th: First IEEE International Workshop on Web Based Systems and Applications(WEBSA)*, 702-707.
- w3schools. (2013). Browser Statistics and Trends.

LAMPIRAN

LAMPIRAN 1

(Kuisisioner *Usability*)

Berikan penilaian mengenai usability dari sistem dengan keterangan sebagai berikut:

- 1 = Sangat tidak setuju
- 2 = Tidak setuju
- 3 = Ragu-ragu
- 4 = Setuju
- 5 = Sangat setuju

NB= - Pastikan semua item/pertanyaan diisi dengan memilih RadioButton sesuai dengan nilai 1-5.

- Pastikan juga bahwa setelah menekan tombol Submit kemudian akan keluar MessageBox "Terima kasih Anda telah memberi masukan pada kami".

- Jika setelah menekan tombol Submit tidak terjadi apa-apa cek ulang kembali setiap item jawaban, untuk memastikan jawaban sesuai dengan keyakinan atau tidak ada pesan error di sisi kanan setiap item. (Setiap user hanya diberi kesempatan 1x pengisian)

		1	2	3	4	5	
1. Secara keseluruhan, saya puas dengan bagaimana mudahnya untuk menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
2. Sederhana untuk menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
3. Saya secara efektif dapat menyelesaikan pekerjaan dengan menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
4. Saya bisa menyelesaikan pekerjaan saya dengan cepat menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
5. Saya bisa menyelesaikan pekerjaan saya dengan efisien menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
6. Saya merasa nyaman menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
7. Mudah untuk belajar menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
8. Saya percaya saya menjadi produktif dengan menggunakan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
9. Aplikasi ini memberikan pesan kesalahan yang jelas dan memberitahu saya cara untuk memperbaiki kesalahan tersebut.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
10. Setiap kali saya melakukan kesalahan dalam menggunakan aplikasi ini, saya dapat kembali dengan mudah dan cepat.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
11. Informasi (seperti bantuan, pada layar pesan, dan dokumentasi lainnya) disediakan aplikasi ini dengan jelas.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
12. Sangat mudah untuk menemukan informasi yang saya butuhkan.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
13. Informasi yang diberikan aplikasi ini mudah untuk dipahami.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
14. Informasi ini efektif dalam membantu saya menyelesaikan tugas.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
15. Informasi di halaman aplikasi jelas.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
16. Antarmuka aplikasi ini menyenangkan.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
17. Saya suka menggunakan antarmuka aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
18. Aplikasi ini memiliki semua fungsi dan kemampuan yang saya harapkan.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju
19. Secara keseluruhan, saya puas dengan aplikasi ini.	Sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat setuju

LAMPIRAN 2

(Development Record)

TRAINING MODULE DEVELOPMENT RECORD

Consultant Name: PT. Mutiara Solusindo	Client Company Name : PT. Schott Igar Glass	Date: 1 January 2013
Module Name: Training	Previously released version number: 2.1 New version number: 2.2	Type of Release: <input type="checkbox"/> ORIGINAL <input type="checkbox"/> MAJORCHANGE <input checked="" type="checkbox"/> MINORCHANGE

DEVELOPMENT LIFECYCLE RECORD			
Activity	Performed	Repaired	Comments
Laporan dari karyawan bahwa email yang diterima tidak mencantumkan lokasi training.	Performed by: Name: Faruq Cahyono Date: 4 Januari 2013	Repaired by: Tim Mutiara 7 Januari 2013	Sistem tidak dapat menampilkan lokasi training pada karyawan.
Uji fungsionalitas	Performed by: Name: Faruq Cahyono Date: 17 Januari 2013	Repaired by: Tim Mutiara 18-21 Januari 2013	Sistem tidak bisa melakukan proses upload pada materi dan sertifikat training.
Uji joba pelaksanaan training	Performed by: Name: Faruq Cahyono Date: 30 Januari 2013	Repaired by: Tim Mutiara 31 Januari 2013	Sistem tidak mengirimkan email pada karyawan sesuai dengan waktu yang telah ditetapkan.
DISTRIBUTION			
Release Date: 4 Februari 2013	Comments: Training Approved Release with new version: 2.2		

LAMPIRAN 3

(Data Responden)

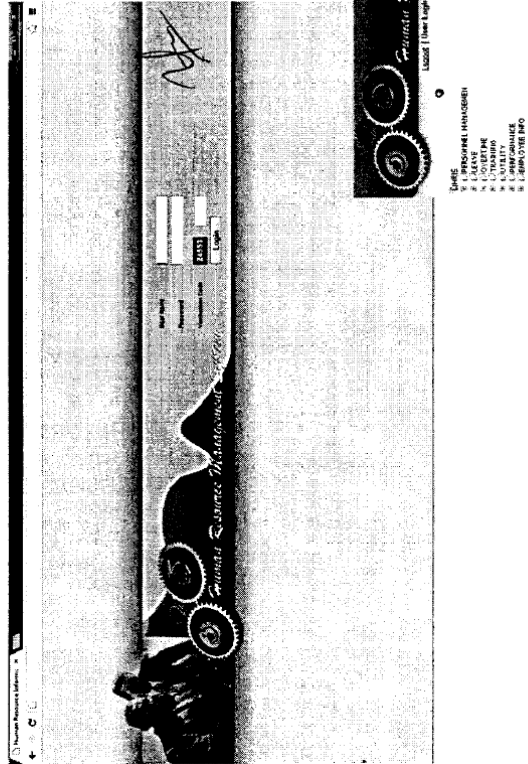
LAMPIRAN 4

User Acceptance Test

(UAT)

UAT DOCUMENTS

Company Name : PT. S [REDACTED] ss
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Login Program
Sub Module Name : User Login
Program Name : Login eHRM



UAT DOCUMENTS

Company Name : PT. S [REDACTED]
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Login Program
Sub Module Name : User Login
Program Name : Login eHRM
Process Name : Entry login program eHRM
Version Access : web
User Category : All User
User Access : All User
User Identity : All User
Sample UAT : Employee_id : 3284
UAT Date :
UAT Revision Date :
Description Process : Pengujian login dengan username dan password
Linked Process :

No	UAT Action	Interface	Pass(P)/Fail(F)		Initials
			User	Manager	
Browser					
1	menampilkan layar login username, password dan verification code	http://10.22.54.6/ehrm		P	NR
Login					
1	isikan NRP karyawan	ketik username		P	NR
2	isikan default password '123'	ketik password		P	NR
3	isikan ulang angka yang tersedia dalam kotak hijau	ketik verification code		P	NR
4	muncul pesan jika login tidak berhasil dan jika berhasil akan tampil jendela awal program eHRM dengan identitas dan sesuai dengan batasan akses menu user yang login	klik tombol login		P	NR

UAT DOCUMENTS

Company Name : PT. Solusi ~~_____~~ **Business**
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Training
Sub Module Name : Master
Program Name : Master Training

Entry Training Master

COMPETENCE PROVIDER TEACHER

Competence Code : General Skill
 Competence Group :
 Required Value :
 Competence Desc :

Provider Name	Type	Address	Teacher
Arbitrade System Informatics	EXTERNAL	Wisma Mega Service, 28th Floor, J. Jendral Sudirman	[0] Person 1
Astra Computer	EXTERNAL	Wisma Jati 2, Jl. Jati Raya No. 7, Gunung Jati	[0] Person 2
All Product Indonesia	EXTERNAL	Building Gedung Babel, Lt. 8, Blok Gedung V-02, 0	[0] Person 3
Akademik Nusantara (Akademik)	EXTERNAL	Wisma Bakti Industri, 11th Floor, Jl. Jendral	[0] Person 4
Akademik Training Center	EXTERNAL	Plaza Center 1 Building 4th Floor, Suite 401, J	[0] Person 5
Akademik Business Consultants	EXTERNAL	Opera House, 115 Scooped	[0] Person 6
Grand Educat Nusantara	EXTERNAL	Merapi Thermo 3rd Floor, Suite 301, Jalan Thermo	[0] Person 7
Dita Social Perdana (Dita Social)	EXTERNAL	Jalisco Ocean Building 3rd Floor, J. Wening Bumi	[0] Person 8

UAT DOCUMENTS

Company Name : PT. Solusi
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Training
Sub Module Name : Master
Program Name : Master Training
Process Name : Entry data master training
Version Access : Web
User Category : Admin User
User Access : HR Staff
User Identity : HR Dept.
Sample UAT : Employee_id : 3284
UAT Date :
UAT Revision Date :
Description Process : Pengujian terhadap proses entry data master training
Linked Process : Portal menu Training Plan

No	UAT Action	Interface	Pass(P)/Fail(F)			Initials
			User	Supervisor	Manager	
Menu						
1	membuka sub menu master	klik menu tree training			P	[Signature]
2	membuka sub menu training	klik menu tree master			P	[Signature]
3	membuka jendela form Entry Training Master	klik menu training			P	[Signature]
Tab Year Period						
1	menampilkan tabel/grid data status plan training per tahun	klik tab year period			P	[Signature]
2	menampilkan form entry data year period	klik tombol add new periode			P	[Signature]
3	menampilkan pesan data sukses tersimpan dan isi data year period training master pada tabel/grid bagian bawah	klik tombol save			P	[Signature]
4	kembali ke tampilan Tab Year Period	klik tombol cancel				
5	menghapus data year period master training dalam tabel/grid	klik tombol delete			P	[Signature]
6	membuka form entry data year period	klik tombol edit			P	[Signature]

7	menampilkan file hasil upload dan file dapat didownload	klik tombol download doc.			P	NA
Tab Competence						
1	menampilkan label/grid data master competence berdasarkan group	klik tab competence			P	NA
2	menampilkan hasil pencarian data sesuai/berdasarkan isian dari data : competence, group dan description	klik tombol search			P	NA
3	menampilkan seluruh data competence dalam tabel/grid	klik tombol reset			P	NA
4	menampilkan form entry data competence	klik tombol new competence			P	NA
5	menampilkan isi data competence pada tabel/grid bagian bawah dan data terurut berdasarkan abjad A - Z.	klik tombol save			P	NA
6	kembali ke tampilan Tab Competence	klik tombol cancel			P	NA
7	menghapus data master competence dalam tabel/grid	klik tombol delete			P	NA
8	membuka form entry data competence dan competence code terkunci	klik tombol edit			P	NA
Tab Provider						
1	menampilkan tabel/grid data master provider	klik tab provide			P	NA
2	menampilkan hasil pencarian data sesuai/berdasarkan isian dari data : nama, address dan city	klik tombol search			P	NA
3	menampilkan seluruh data provider dalam tabel/grid	klik tombol reset			P	NA
4	menampilkan form entry data provider	klik tombol new provider			P	NA
5	menampilkan isi data provider pada tabel/grid bagian bawah	klik tombol save			P	NA
6	kembali ke tampilan Tab Provider	klik tombol cancel			P	NA
7	menghapus data provide dalam tabel/grid	klik tombol delete			P	NA
8	membuka form entry data provider	klik tombol edit			P	NA
9	menampilkan tabel/grid data trainer	klik tombol person			P	NA
10	kembali ke tampilan Tab Provider	klik tombol back provider			P	NA
11	menampilkan form entry data trainer	klik tombol add trainer			P	NA
12	untuk menyimpan data trainer dalam tabel/grid	klik tombol add			P	NA
13	kembali ke tampilan tabel/grid data trainer	klik tombol cancel			P	NA
Other Function						
1	melihat keseluruhan data yang ada	scroll atas atau bawah			P	NA

UAT DOCUMENTS

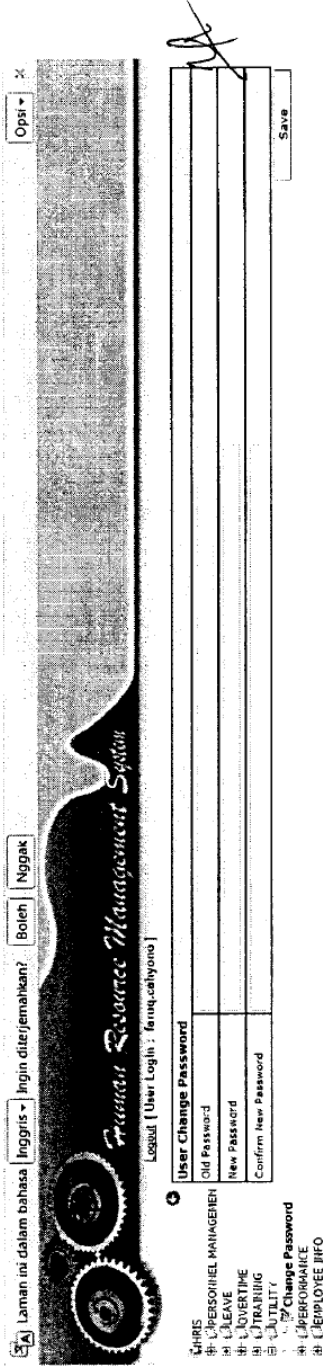
Company Name : PT. Solusi XXXXXXXXXX **Pass**
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Training
Sub Module Name : Master
Program Name : Master Topic
Process Name : Entry data master Topic
Version Access : Web
User Category : Admin User
User Access : HR Staff
User Identity : HR Dept.
Sample UAT : Employee_id : 3284
UAT Date :
UAT Revision Date :
Description Process : Pengujian terhadap proses entry data master topic training
Linked Process : Training Plan

No	UAT Action	Interface	Pass(P)/Fail(F)		Initials
			User	Supervisor / Manager	
Menu					
1	membuka sub menu master	klik menu tree training		P	ef
2	membuka sub menu Topic	klik menu tree master		P	ef
3	membuka jendela form Entry Training Topic	klik menu Topic		P	ef
Form Entry Training Topic					
1	menampilkan hasil pencarian data sesuai/berdasarkan isian dari data : subject, description dan training_id	klik tombol search		P	ef
2	menampilkan seluruh data topic dalam tabel/grid	klik tombol reset		P	ef
3	menampilkan form Add Topic	klik tombol add new topic		P	ef
4	menampilkan isi data topic training pada tabel/grid bagian bawah	klik tombol save		P	ef
5	kembali ke tampilan Entry Training Topic	klik tombol cancel		P	ef
6	menghapus data training topic dalam tabel/grid	klik tombol delete		P	ef
7	membuka form Add Topic	klik tombol edit		P	ef

Kolom Other Document				
8	menampilkan file hasil upload dan file dapat didownload	klik tombol view doc.	P	✓
Kolom Competence				
9	menampilkan form data competence	Klik tombol item	P	✓
10	menampilkan form entry data competence	Klik tombol add new competence	P	✓
11	unituk kembali ke tampilan Entry Training Topic	Klik tombol back	P	✓
Kolom Prerequisite				
12	menampilkan tabel/grid prerequisite	Klik tombol item	P	✓
13	menampilkan form entry data prerequisite	Klik tombol add new prerequisite	P	✓
14	menampilkan isi data prerequisite topic pada tabel/grid bagian bawah	Klik tombol add	P	✓
15	kembali ke tampilan tabel/grid prerequisite	Klik tombol cancel	P	✓
16	unituk kembali ke tampilan Entry Training Topic	Klik tombol back	P	✓
17	menghapus data prerequisite topic dalam tabel/grid	Klik tombol delete	P	✓
Other Function				
1	melihat keseluruhan data yang ada	scroll atas atau bawah	P	✓

UAT DOCUMENTS

Company Name : PT. Solusi ~~_____~~ **Business**
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Utility
Sub Module Name :
Program Name : Utility



UAT DOCUMENTS

Company Name : PT. S[REDACTED] ss
Consultant Name : PT. Mutiara Solusindo
Program Name : SolutionPRO E-HRM Professional
Module Name : Utility
Sub Module Name :
Program Name : Utility
Process Name : Entry perubahan password
Version Access : Web
User Category : All User
User Access : All User
User Identity : All User
Sample UAT : Employee_id : 3284
UAT Date :
UAT Revision Date :
Description Process : Pengujian terhadap entry perubahan password user
Linked Process :

No	UAT Action	Interface	Pass(P)/Fail(F)		Initials
			User	Supervisor / Manager	
Menu					
1	membuka sub menu change password	klik tree menu Utility		P	[Signature]
2	membuka jendela user change password	klik menu change password		P	[Signature]
User Change Password					
1	password lama akan tergantikan dengan password baru ketika user melakukan login ulang	klik tombol save		P	[Signature]

LAMPIRAN 5

(Surat Keputusan Pembimbing)

**KEPUTUSAN DEKAN
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : 179/ELK/Q-I/X/2012
TENTANG**

**PENGANGKATAN PEMBIMBING TUGAS AKHIR SKRIPSI
BAGI MAHASISWA FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

**DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

- Menimbang** : 1. Bahwa sehubungan dengan telah dipenuhi syarat untuk penulisan Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta, perlu diangkat pembimbing.
2. Bahwa untuk keperluan dimaksud perlu ditetapkan dengan Keputusan Dekan.
- Mengingat** : 1. Undang-undang Nomor 20 tahun 2003.
2. Peraturan Pemerintah RI Nomor 60 tahun 1999.
3. Keputusan Presiden RI: a. Nomor 93 tahun 1999; b. 305/M tahun 1999.
4. Keputusan Menteri Pendidikan dan Kebudayaan RI: Nomor 274/O/1999.
5. Keputusan Mendiknas RI Nomor 003/O/2001.
6. Keputusan Rektor UNY Nomor : 1160/UN34/KP/2011

MEMUTUSKAN

Menetapkan

Pertama : Mengangkat Pembimbing Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta sebagai berikut :

Nama Pembimbing : Handaru Jati, Ph.D
Bagi mahasiswa :
Nama/No.Mahasiswa : Imanaji Hari Sayekti / 09520244011
Jurusan/ Prodi : Pendidikan Teknik Elektronika / Pendidikan Teknik Informatika


Kedua : Dosen pembimbing disertai tugas membimbing penulisan Tugas Akhir Skripsi sesuai dengan Pedoman Tugas Akhir Skripsi.

Ketiga : Keputusan ini berlaku sejak ditetapkan

Keempat : Segala sesuatu akan diubah dan dibetulkan sebagaimana mestinya apabila di kemudian hari ternyata terdapat kekeliruan dalam Keputusan ini.

Ditetapkan : di Yogyakarta
Pada tanggal : 16 Oktober 2012

Dekan


Dr. Moch./Bruri Triyono
NIP. 19560216 198603 1 003

Tembusan Yth :

1. Wakil Dekan II FT UNY
2. Ketua Jurusan Pendidikan Teknik Elektronika
3. Kasub. Bag Pendidikan FT UNY
4. Yang bersangkutan

LAMPIRAN 6

(Keterangan Selesai Penelitian)



PT MUTIARA SOLUSINDO

Jl. Rawa Selatan V No.6 Johar Baru, Jakarta Pusat – Indonesia
Phone: (+62-21) 424 6092 Fax: (+62-21) 424 1989
EMail: info@mutisindo.com WebSite: www.mutisindo.com

SURAT KETERANGAN
017/MS/02/2013/SKP

Kami selaku HR Direktur **PT. Mutiara Solusindo** yang bergerak di bidang *Consultant Management and IT*, menerangkan dengan sesungguhnya bahwa yang bersangkutan di bawah ini:

Nama : **Imanaji Hari Sayekti**
Nama Universitas : Universitas Negeri Yogyakarta
Program Studi : Pendidikan Teknik Informatika
Nim Mahasiswa : 09520244011

Telah melaksanakan penelitian skripsi di perusahaan kami, PT. Mutiara Solusindo dengan judul **“ANALISIS PENGEMBANGAN SISTEM APLIKASI *E-TRAINING* KARYAWAN BERBASIS *WEB* PADA PT. MUTIARA SOLUSINDO”** selama 3 (tiga) bulan. Terhitung dari tanggal 1 November 2012 sampai dengan 1 Februari 2013.

Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Jakarta, 4 Februari 2013

Hormat kami,

Ir. Hendrik Braja
HR Direktur