

**APLIKASI PENILAIAN RESIKO PENGEMBANGAN PERANGKAT LUNAK DI
PERGURUAN TINGGI BERDASARKAN SOFTWARE ENGINEERING RISK
MANAGEMENT(MODUL: FASE PENGEMBANGAN, AKTIVITAS RESIKO
PENGEMBANGAN PERANGKAT LUNAK, LAPORAN)**

Rachma Muslimatunnisa¹, Sanza Vittria Wulanda², Heru Nugroho³, Ferra Arik Tridalestaria⁴

123 Program Studi D3 Manajemen Informatika, Fakultas Ilmu Terapan, Universitas Telkom

rachma.muslimatun@gmail.com.

Abstrak

Penelitian ini dilatar belakangi oleh banyaknya perguruan tinggi yang menginginkan proyek perangkat lunaknya dinilai untuk mengukur keberhasilan proyek yang dimiliki perguruan tinggi tersebut. Perkembangan teknologi informasi khususnya perangkat lunak yang merupakan proses dan melibatkan banyak pihak serta memiliki ketidakpastian sukses yang cukup tinggi. Penilaian dilakukan dengan melakukan penyebaran kuisioner. Saat ini perhitungan dan penginputan data masih di lakukan secara manual di Microsoft Excel, sehingga mengakibatkan adanya kemungkinan salah input data ataupun salah perhitungan. Jika data yang di hasilkan tidaklah konkrit maka mutu proyek yang dimiliki pun semakin menurun. Jika data yang dihasilkan konkrit, maka mutu sebuah proyek tersebut juga akan semakin baik. Adapun tujuan dari penelitian ini adalah sebagai berikut : untuk membuat sebuah aplikasi yang bisa menghasilkan sebuah angka yang mengukur bagusnya sebuah proyek perangkat lunak yang bernama Software Engineering Risk Management.

Dalam penelitian ini penulis menggunakan jenis penelitian deskriptif yang dimana penelitian deskriptif bertujuan untuk memberikan atau menjabarkan suatu keadaan atau fenomena yang terjadi. Sedangkan penelitian kausal adalah penelitian yang menjelaskan hubungan sebab akibat antar variabel.

Kata kunci: Rencana Kerja SD Ar-Rafi', Rencana Kerja Tahunan, Pemantauan Rencana Kerja.

Abstract

This research was motivated by the many colleges that want their software projects assessed to measure the success of a project owned by the college. The development of information technology, especially software which is a process involving many parties as well as the uncertainty of success are quite high. This time calculation and data entry was done manually in Microsoft Excel, resulting in the possibility of incorrect data input or miscalculation. If the data that produced the concrete is not owned any project quality decreases. If the data generated concrete, the quality of a project will also be getting better.

The purpose of this study are as follows: to create an application that can generate a good number that measures a software project called Software Engineering risk management.

In this study the authors menggunakan descriptive research which aims to provide a descriptive study or describe a situation or phenomenon that occurs. While kausal research is research that explains the causal relationship between variables.

Keyword: Quetioner, Assesment, Higher Education, Software Engineering Risk Management.

1. Pendahuluan

1.1 Latar Belakang

Pengembangan teknologi informasi khususnya perangkat lunak merupakan proses yang melibatkan banyak pihak seperti klien dan *stakeholder*(*Chief Information Officer, Senior IT, Project Manager, System Analyst, Bussiness Analyst, Database Administrator, Programmer*) dan memiliki ketidakpastian sukses yang cukup tinggi. Sehingga jika tidak dikelola dengan baik, maka akan terjadi ketidaksesuaian harga dengan hasil yang dicapai. Menurut Sudono (2010), dalam berbagai survey yang telah dilakukan, rata-rata angka kegagalan proyek teknologi informasi mencapai 70%. Imam dan Totok (2015) menyatakan bahwa kenyataan yang terjadi di lapangan sebagian besar (sekitar 90%) proyek tak pernah tuntas. Selain itu, Rudy Tantra (2012) menyatakan bahwa ada enam faktor yang menyebabkan kegagalan proyek. Beberapa penyebab kegagalan proyek yaitu:

1. Kegagalan perencanaan
Kegagalan ini terjadi jika proyek tidak memiliki nilai tambah atau kemampuan untuk menghadapi halangan yang terjadi sepanjang prosesnya.
2. Kegagalan sponsor
Kegagalan ini terjadi ketika orang yang memiliki proyek tidak secara aktif terlibat dan/atau tidak memiliki otoritas untuk mengambil keputusan penting demi kesuksesan proyek.
3. Kegagalan desain dan definisi ruang lingkup
Kegagalan ini terjadi jika ruang lingkup tidak didefinisikan dengan jelas sehingga tim pelaksana proyek tidak memahami dengan jelas apa yang menjadi *deliverables*.
4. Kegagalan komunikasi

Kegagalan ini terjadi karena kurangnya komunikasi atau menghindari diskusi terbuka mengenai masalah dan isu yang terjadi di dalam proyek.

5. Kegagalan disiplin proyek
Kegagalan ini terjadi ketika proses atau metodologi proyek mengabaikan kesalahan kecil sehingga faktor mitigasi yang tidak pernah digunakan.

6. Kegagalan pemasok/*vendor*
Kegagalan ini terjadi jika *vendor* tidak dapat berkomunikasi secara langsung dengan tim pelaksana proyek karena faktor birokrasi perusahaan.

Dalam pengembangan proyek perangkat lunak seringkali terdapat resiko-resiko yang tak terduga. Sehingga dapat menyebabkan kendala dalam pengerjaan proyek. Seorang manajer proyek harus bisa mengukur tingkat keberhasilan proyek. Untuk mengetahui tingkat keberhasilan proyek, maka perlu diketahui sejak dini letak resiko yang akan menyebabkan kegagalan proyek.

Dalam pengembangan perangkat lunak di perguruan tinggi, resiko-resiko tersebut sebenarnya dapat dideteksi sejak dini jika kita menggunakan konsep manajemen resiko dan mengimplementasikan ke dalam proyek pengembangan perangkat lunak. Jika letak resiko sudah diketahui, maka kita dapat menyiapkan beberapa solusi untuk mengurangi resiko dan dampak negatifnya. *Software Engineering Risk Management (SERIM)* merupakan *tools* yang digunakan untuk memecahkan pemecahan alternatif resiko pada suatu proyek perangkat lunak.

Penggunaan SERIM untuk menilai pengembangan perangkat lunak saat ini masih menggunakan *Microsoft Excel*. Dalam proses pengisian form penilaian dan perhitungan hasil survey yang dilakukan oleh user masih dilakukan secara manual. Sehingga ada beberapa data yang memungkinkan adanya kesalahan saat memasukkan nilai. Selain itu, pengisian kuisisioner harus dilakukan di tempat responden bekerja. Permasalahan lain yang muncul adalah terkadang pengerjaan pengisian kuisisioner SERIM tidak dapat diselesaikan pada satu waktu tertentu. Hasil dari SERIM untuk menghitung pengembangan perangkat lunak sangat banyak dan rumit.

Proses perhitungan tersebut dengan menentukan peluang dari *Risk Factor (Organization, Estimation, Monitoring, Development Methodology, Tools, Risk Culture, Usability, Correctness, Reliability, dan Personel)*. Penentuan peluang berdasarkan nilai yang telah diisi pada kuisisioner yang dilakukan oleh responden. Hasil dari *Risk factor* akan menentukan peluang dari unsur-unsur yaitu:

1. *Risk Element* (Modul A)
2. *Total Product Risk* (Modul A)

3. *Development Phases*
4. *Software Management Risk Activities*

Development Phases terdiri dari:

1. *Pre-Requirements*
2. *Requirements*
3. *Design*
4. *Code*
5. *Test*
6. *Development and maintenance*

Software Management Risk Activities terdiri dari:

1. *Identification*
2. *Strategy and Planning*
3. *Assesment*
4. *Mitigation and Avoidance*
5. *Reporting*
6. *Prediction*
7. *Process*
8. *Product*

Hasil dari *Development Phases* dan *Software Risk Management Activities* didapatkan dari beberapa unsur dari *Risk Factor* yang kemudian dikalikan dengan bobot yang telah ditentukan. Dari nilai *Risk Factor, Risk Element, Total Product Risk, Development Phases*, dan *Software Risk Management Activities* akan menghasilkan laporan berupa deskripsi dari peluang terkecil. Dari hasil penilaian inilah akan diketahui letak resiko dari proyek akan dijalankan.

Dalam pembuatan aplikasi akan dibagi menjadi dua modul. Tanpa akan mengerjakan modul A yang terdiri dari *Risk Factor, Risk Element, Total Product Risk*. Sedangkan Rachma mengerjakan modul B yang terdiri dari *Development Phases, Software Risk Management Activities*, dan laporan. Untuk pembuatan aplikasi dari mulai login hingga penginputan kuisisioner akan dilakukan bersama-sama.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam proyek akhir ini adalah :

1. Bagaimana membangun aplikasi berbasis web yang dapat memberikan penilaian terhadap resiko pengembangan perangkat lunak?
2. Bagaimana membuat kuisisioner resiko TI yang dapat diakses user secara online?
3. Bagaimana membangun aplikasi yang memiliki fitur penyimpanan sementara dari responden?
4. Bagaimana menghitung peluang faktor resiko dari unsur *Development Phases*?

5. Bagaimana menghitung peluang faktor resiko dari unsur *Software Risk Management Activities*?
6. Bagaimana membuat deskripsi dalam bentuk laporan berdasarkan hasil pada *Risk Factor, Risk Element, Total Product Risk (Modul A), Software Risk Management Activities, dan Development Phases*?

1.3 Tujuan

Tujuan dari proyek akhir ini adalah membangun Aplikasi Penilaian Resiko Teknologi Informasi di Perguruan Tinggi Berdasarkan *Software Engineering Risk Management (SERIM)* yang memiliki fitur:

1. Input penilaian terhadap resiko pembangunan perangkat lunak.
2. Menggunakan aplikasi input penilaian pembangunan resiko perangkat lunak.
3. Penyimpanan jawaban dari responden untuk setiap penilaian resiko TI.
4. Menghitung penilaian keberhasilan aktivitas resiko perangkat lunak terhadap unsur *Development Phases*.
5. Menghitung penilaian keberhasilan aktivitas resiko perangkat lunak terhadap unsur *Software Risk Management Activities*.
6. Membuat deskripsi laporan produk perangkat lunak yang didapat dari hasil perhitungan peluang dari *Risk Factor, Risk Element, Total Product Risk (Modul A), Software Risk Management Activities, dan Development Phases*.

1.4 Batasan Masalah

Adapun batasan masalah dalam pembuatan proyek akhir ini yaitu:

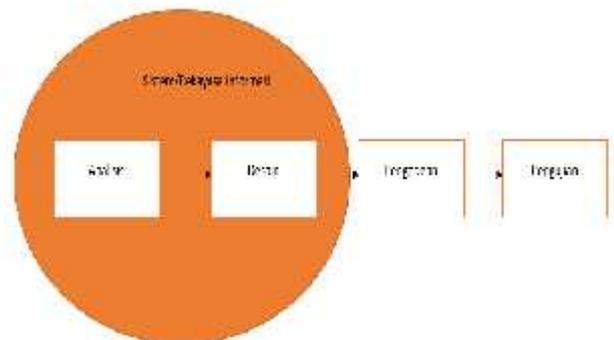
1. Deskripsi laporan berupa hasil perhitungan peluang terkecil dari unsur *risk factor, risk element, development phases, software risk management activities, dan total product risk*.
2. Tahapan pengembangan (metode pengerjaan) hanya dilakukan sampai tahap implementasi, tidak sampai pada tahap *operation*.
3. Penggunaan aplikasi hanya dilakukan untuk Perguruan Tinggi yang melakukan investasi TI.

1.5 Definisi Operasional

Aplikasi Penilaian Resiko Teknologi Informasi di Perguruan Tinggi Berdasarkan *Software Engineering Risk Management* adalah aplikasi yang dirancang untuk memberikan estimasi peluang keberhasilan sebuah proyek pengembangan perangkat lunak perguruan tinggi. Aplikasi dapat menghitung peluang *Risk Factor, Risk element, Development Phases, Software Risk Management Activities, dan Total Product Risk*. Aplikasi berbasis web ini menggunakan bahasa pemrograman PHP, HTML, dan CSS. Pengerjaan aplikasi ini menggunakan metode *waterfall* dengan tahap analisis kebutuhan, desain, pembuatan kode, pengujian, dan implementasi. Aplikasi ini menggunakan database server berupa *MySQL* karena kecepatan dan ukurannya yang kecil membuatnya ideal untuk website. Aplikasi ini diharapkan dapat membantu perguruan tinggi dalam menentukan peluang keberhasilan terhadap pembangunan perangkat lunak secara keseluruhan.

1.6 Metode Pengerjaan

Metodologi yang dipergunakan dalam pengerjaan proyek akhir ini adalah tahapan model *waterfall*. Berikut adalah beberapa tahapan dari *waterfall*, yaitu:



Gambar 1-1 Schema Waterfall Presman[1].

- a. Analisis: Pada tahap ini penulis mencari referensi dan tools SERIM yang berkaitan dengan aplikasi untuk menghitung resiko TI. Salah satu yang menjadi referensi adalah Aplikasi Tata Kelola Teknologi Informasi berdasarkan Kerangka Kerja COBIT 4.1 (Modul: IT Audit). Penulis juga mempelajari perhitungan resiko TI, menggunakan SERIM versi Excel.
- b. Desain: Setelah mendapatkan informasi tentang pentingnya aplikasi untuk mengetahui tingkat resiko pengembangan perangkat lunak maka penulis membuat perancangan basis data dengan *Entity Relation Diagram (ERD)*, merancang *Use*

case yang merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat, merancang *flow map* untuk menggambarkan proses bisnis berjalan dan usulan, dan merancang tampilan aplikasi dengan menggunakan *balsamiq mockups*.

- c. Pengodean: Pada tahap ini penulis menerjemahkan desain system yang telah dibuat kedalam bahasa pemrograman *PHP* dan database server berupa *MySQL* yang nantinya akan menjadi sebuah aplikasi yang dapat diakses oleh responden perguruan tinggi.
- d. Pengujian: Pada tahap ini akan dilakukan pengujian system menggunakan metode black box dan *User Acceptance Test (UAT)* untuk memastikan bahwa aplikasi yang dibangun telah memenuhi dari responden Perguruan Tinggi.

2. Tinjau Pustaka

2.1 Manajemen Resiko Teknologi Informasi

Manajemen resiko TI merupakan suatu bentuk pengakuan akan adanya ancaman, menentukan konsekuensinya terhadap sumber daya, serta menerapkan modifikasi faktor-faktor biaya yang telah dibuat untuk menjaga adanya konsekuensi yang tidak diharapkan. Tujuan dari manajemen resiko TI adalah menjaga aset-aset TI seperti data, perangkat keras, perangkat lunak, sumber daya manusia, serta infrastruktur[3]. Analisis resiko pada dasarnya adalah proses yang diadaptasi dari praktik manajemen dan merupakan suatu pendekatan untuk setiap upaya kedepan dalam rangka menetapkan keamanan dari teknologi informasi. Isu utama yang mencakup di dalamnya adalah kebutuhan metode sistematis untuk mengidentifikasi keterkaitan antar resiko TI dan keamanan total dari TI sulit untuk dilakukan[4].

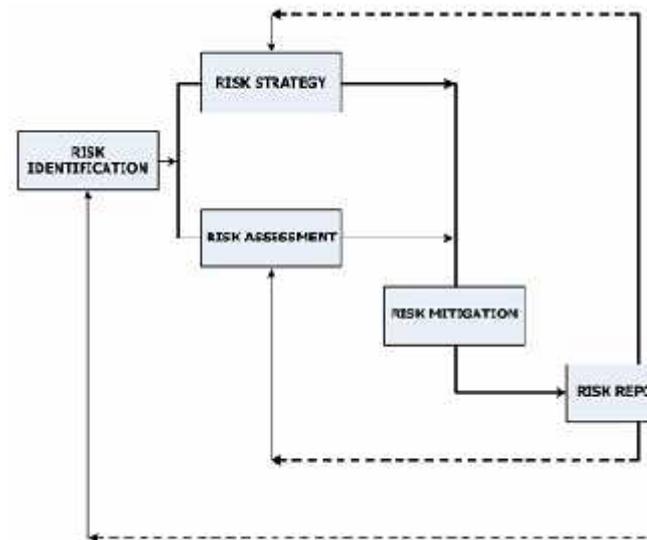
Manajemen resiko perangkat lunak adalah pengelolaan dan minimasi kegagalan yang mencakup aspek fungsionalitas, *cost overruns*, dan *schedule overruns* pada pengembangan perangkat lunak (Karolak, 1999). Tiga area pokok dari resiko pengembangan perangkat lunak tersebut dijabarkan sebagai berikut[5]:

- a. Tidak adanya kejelasan akan kebutuhan perangkat lunak sehingga mengakibatkan ketidaktepatan fungsionalitas yang dikembangkan.
- b. Ketidapahaman dalam estimasi biaya yang akan digunakan dalam mengembangkan perangkat lunak sehingga mengakibatkan biaya berlebihan.
- c. Ketidakmampuan dalam mengukur kinerja tim pengembang perangkat lunak dalam

menyelesaikan pekerjaannya dan besarnya fungsionalitas sehingga mengakibatkan pemuluran jadwal pengembangan perangkat lunak tersebut. Kegiatan yang dilakukan dalam manajemen resiko (Karolak, 1999) adalah:

- a) *Risk Identification*, yaitu melakukan identifikasi gejala resiko yang terjadi.
- b) *Risk Strategy*, yaitu merancang suatu tahapan langkah untuk menanggulangi resiko.
- c) *Risk Assessment*, yaitu mengukur akibat yang akan disebabkan resiko.
- d) *Risk Mitigation*, yaitu melakukan mitigasi dari hasil penilaian resiko.
- e) *Risk Reporting*, yaitu membuat penulisan terhadap seluruh kegiatan manajemen resiko sehingga dapat digunakan sebagai dasar analisis manajemen resiko berikutnya.
- f) *Risk Prediction*, yaitu membuat perkiraan akan resiko yang akan terjadi berikutnya dalam pengembangan perangkat lunak.

Analisis Manajemen Resiko Perangkat Lunak dengan Pendekatan Just-in-Time: Studi Kasus Optimasi Organisasi dan Dokumentasi pada Organisasi Pengembang Perangkat Lunak (Thomas Suselo)



Gambar 2-1 Aktivitas Manajemen Resiko JIT Perangkat Lunak[5].

Pada penelitian ini yang akan dikaji adalah seberapa besar kuantitas resiko pada kasus organisasi pengembang perangkat lunak yaitu:

- a) Tidak membuat dokumentasi pembangunan perangkat lunak (dokumen Spesifikasi Kebutuhan Perangkat Lunak/ SKPL), dokumentasi user-manual, dan penetapan dokumentasi tersebut menjadi standar dalam melakukan pengembangan perangkat lunak.

- b) Manajemen organisasi tidak dapat mengkomunikasikan anggota pengembang dengan baik sehingga setiap anggota cenderung berjalan sendiri-sendiri. Kuantitas resiko tersebut haruslah diminimasi dengan menggunakan manajemen resiko. Pendekatan yang dilakukan terhadap manajemen resiko kasus ini adalah *Just-In Time*(JIT) dengan tools *Software Engineering Risk Model*(SERIM).

2.2 Software Engineering Risk Management(SERIM)

Karolak (1999) meneliti suatu model yang dapat digunakan sebagai acuan manajemen resiko dengan pendekatan JIT yang disebut sebagai Software Engineering Risk Model (SERIM). Model tersebut merupakan model probabilitas yang mengakomodasikan elemen-elemen berikut:

- a. *Technical Risk* terdiri atas aspek-aspek *functionality, quality, reability, usability, timelines, maintainability, reusability*.
- b. *Cost Risk*, terdiri atas aspek-aspek *budget, nonrecurring cost, recurring cost, fixed cost, variable cost*.
- c. *Schedule Risk*, terdiri atas aspek-aspek *flexibility, Meeting Established Milestones, Realism*.

Pada SERIM, aspek-aspek pada tiap elemen diatas diterjemahkan menjadi 10 faktor resiko sebagai berikut:

- a) *Organization*
- b) *Estimation*
- c) *Monitoring*
- d) *Development methodology*
- e) *Tools*
- f) *Risk Culture*
- g) *Usability*
- h) *Correctness*
- i) *Reliability*
- j) *Personel*

Faktor resiko inilah yang kemudian diukur dalam *risk metrics* yang diformulasikan ke dalam 81 pertanyaan. *Risk Metrics* pada SERIM menggunakan konsep pohon probabilitas yang menunjukkan muatan resiko sebagai rujukan untuk antisipasi atapun pengembangan produk perangkat lunak, atau bahkan kinerja organisasi tersebut. Alur kalkulasi

rentang nilai pada pohon probabilitas mencerminkan formulasi faktor resiko yang dihadapi organisasi (tertuang dalam 81 pertanyaan). Masing-masing pertanyaan dalam *risk metrics* dijawab (secara *self-assessment*) dengan nilai rentang 0-1, hal tersebut bertujuan untuk membangun satuan probabilitas pengembangan proses. Satuan probabilitas kemudian dikelompokkan menurut aktivitas manajemen resiko, tahapan pengembangan, dan faktor resiko. Faktor resiko kemudian dikelompokkan berdasarkan elemen-elemen resiko untuk kemudian dipadukan sehingga menghasilkan total resiko pengembangan perangkat lunak[5].

2.3 Pendekatan Just-In-Time (JIT)

Konsep JIT pada pengembangan perangkat lunak, filosofinya bertumpu pada fungsionalitas, biaya dan waktu (jadwal). Manajemen organisasi perangkat lunak kadangkali memandang proses pengembangan perangkat lunak sebagai proses yang sangat tidak dapat digambarkan (abstrak), sehingga tidak didapatkan pengukuran fungsionalitas yang dibutuhkan. Tahap awal inilah yang memicu *cost-overruns* dan *schedule-overruns*. JIT pada pengembangan perangkat lunak merupakan pendekatan yang dilakukan oleh pihak manajemen organisasi yang bersifat *risk-driven*, dimana konsepnya adalah sebagai berikut:

- a. Antisipasi dan minimasi resiko dalam pengembangan perangkat lunak.
- b. Menangani resiko sejak dini dalam pengembangan perangkat lunak sehingga mengurangi waktu siklus proses, yang akan berimbas pada pengurangan biaya, pemenuhan jadwal, dan kesesuaian fungsionalitas. Dalam hal melakukan manajemen resiko perlu untuk memahami dan mengakomodasi semua perspektif sebagai berikut dan perspektif tersebut akan dijadikan dasar untuk melakukan kegiatan manajemen resiko, yang telah diuraikan pada pembahasan mengenai pengertian manajemen resiko[5]:
 - a) Operasional: berkaitan dengan ketidakpastian dalam kegiatan rutin-harian.
 - b) Strategis: berkaitan dengan dampak jangka panjang bagi organisasi/perusahaan.
 - c) Teknis: berkaitan dengan penggunaan teknologi perangkat lunak.
 - d) Bisnis: berkaitan dengan proyek-proyek yang dilakukan organisasi/perusahaan dalam berbagai bentuk komersialitasnya.

- e) Industri: berkaitan dengan model dan proses pengembangan perangkat lunak yang berbasis industri (definitif, terkuantifikasi, sistematis).
- f) Praktisi: berkaitan dengan implementasi dan praktik-praktik pengembangan perangkat lunak.

Model Persamaan

Untuk menerapkan model di angka 9-2 dan 9-3, beberapa parameter, eters dan equations harus mengidentifikasi dipertimbangkan. Persamaan berikut digunakan untuk masing-masing pohon probabilitas :

1. $P(A) = [\sum_{n=1}^3 P(A_n)]/3$
diasumsikan bahwa setiap elemen memiliki bobot yang sama. Jika bobot masing-masing elemen berbeda di antara mereka, maka $P(A) = w_1P(A_1) + w_2P(A_2) + w_3P(A_3)$ dimana setiap, adalah bilangan positif dan $w_1, w_1 + w_2 + w_3 = 1$
2. $P(A_1) = [\sum_{n=1}^1 w_n P(A_n)]$ dimana $w_4 = 0.043, w_5 = 0.043, w_6 = 0.087, w_7 = 0.087, w_8 = 0.087, w_9 = 0.13, w_{10} = 0.13, w_{11} = 0.13, w_{12} = 0.13, w_{13} = 0.13, w_{14} = 0.13$. Berdasarkan tabel 6-1, bobot 0.043 diperuntukan untuk nilai yang rendah, 0.087 untuk nilai menengah, dan 0.13 untuk nilai tinggi.
3. $P(A_2) = [\sum_{n=1}^1 w_n P(A_n)]$ dimana $w_4 = 0.136, w_5 = 0.136, w_6 = 0.136, w_7 = 0.136, w_8 = 0.09, w_9 = 0.09, w_{10} = 0.045, w_{11} = 0.045, w_{12} = 0.045, w_{13} = 0.136$. Berdasarkan tabel 6-1, bobot 0.045 diperuntukan untuk nilai yang rendah, 0.09 untuk nilai menengah, dan 0.136 untuk nilai tinggi.
4. $P(A_3) = [\sum_{n=1}^1 w_n P(A_n)]$ dimana $w_4 = 0.136, w_5 = 0.136, w_6 = 0.136, w_7 = 0.136, w_8 = 0.09, w_9 = 0.09, w_{10} = 0.045, w_{11} = 0.045, w_{12} = 0.045, w_{13} = 0.136$. Berdasarkan tabel 6-1, bobot 0.045 diperuntukan untuk nilai yang rendah, 0.09 untuk nilai menengah, dan 0.136 untuk nilai tinggi.
5. $P(A_4) = [\sum_{n=1}^8 (O)] / 8$ di mana O adalah nilai metrik untuk jumlah pertanyaan O yang diidentifikasi dalam point 7
6. $P(A_5) = [\sum_{n=1}^7 (E)] / 7$ di mana E adalah nilai metrik untuk jumlah pertanyaan E yang diidentifikasi dalam point 7

7. $P(A_6) = [\sum_{n=1}^7 (M)] / 7$ di mana M adalah nilai metrik untuk jumlah pertanyaan M yang diidentifikasi dalam point 7
8. $P(A_7) = [\sum_{n=1}^7 (DM)] / 7$ di mana DM adalah nilai metrik untuk jumlah pertanyaan DM yang diidentifikasi dalam point 7
9. $P(A_8) = [\sum_{n=1}^9 (T)] / 9$ di mana T adalah nilai metrik untuk jumlah pertanyaan T yang diidentifikasi dalam point 7
10. $P(A_9) = [\sum_{n=1}^7 (M)] / 7$ di mana M adalah nilai metrik untuk jumlah pertanyaan M yang diidentifikasi dalam point 7
11. $P(A_{10}) = [\sum_{n=1}^6 (U)] / 6$ di mana U adalah nilai metrik untuk jumlah pertanyaan U yang diidentifikasi dalam point 7
12. $P(A_{11}) = [\sum_{n=1}^9 (C)] / 9$ di mana C adalah nilai metrik untuk jumlah pertanyaan C yang diidentifikasi dalam point 7
13. $P(A_{12}) = [\sum_{n=1}^1 (R)] / 12$ di mana R adalah nilai metrik untuk jumlah pertanyaan R yang diidentifikasi dalam point 7
14. $P(A_{13}) = [\sum_{n=1}^5 (P)] / 5$ di mana P adalah nilai metrik untuk jumlah pertanyaan P yang diidentifikasi dalam point 7
15. $P(B) = (O1, O2, O3, O4, O5, E1, E2, E3, E4, E6, E7, M1, M2, M3, M4, M6, M7, DM1, DM2, DM6, T1, T6, T9, RC1, RC2, RC3, RC4, RC5, RC6, RC7, RC8, RC9, RC10, RC11, C5, P1, P2, P3, P4, P5) / 40$
16. $P(C) = (O1, O3, O4, O5, O6, O7, O8, E5, E6, E7, M3, M4, M5, M6, M7, DM2, DM3, DM4, DM6, T1, T6, T9, RC1, RC2, RC3, RC4, RC5, RC6, U2, U3, U5, C1, C5, R3) / 34$
17. $P(D) = (O1, O3, O4, O5, O6, O7, O8, E5, E6, E7, M3, M4, M5, M6, M7, DM2, DM3, DM4, DM6, T1, T2, T6, T9, RC1, RC2, RC3, RC4, RC5, RC6, U2, U4, U6, C2, C5, R1, R2, R3, R5) / 38$
18. $P(E) = (O1, O3, O4, O5, O6, O7, O8, E5, E6, E7, M3, M4, M5, M6, M7, DM2, DM3, DM4, DM6, T1, T6, T8, T9, RC1, RC2, RC3, RC4, RC5, RC6, U2, C3, C5, C6, R1, R2, R3, R4, R5, R6) / 39$
19. $P(F) = (O1, O3, O4, O5, O6, O7, O8, E5, E6, E7, M3, M4, M5, M6, M7, DM2, DM3, DM5, DM6, DM7, T1, T3, T4, T5,$

T6, T9, RC1, RC2, RC3, RC4, RC5, RC6, U2, C4, C5, C7, C9, R7, R9, R10, R11, R12)/42

20. $P(G) = (O1, O3, O4, O5, O6, O7, O8, E5, E6, E7, M3, M4, M5, M6, M7, DM2, DM3, DM6, T1, T6, T7, T9, RC1, RC2, RC3, RC4, RC5, RC6, U1, U2, C5, C8, R8)/33$
21. $P(H) = (\sum_{n=1}^8 Q_n)/81$ dimana Q_n adalah pertanyaan metrik dari (O1...O8, E1...E7, M1...M7, DM1...DM7, T1...T9, RC1...RC11, U1...U6, C1...C9, R1...R12, P1...P5)
22. $P(I) = (O3, E1, E2, E3, E4, E5, E6, E7, M1, DM1, RC11, P1)/12$
23. $P(J) = (O1, O2, O3, O7, O8, E2, E3, E4, E5, E6, E7, M5, M6, M7, DM3, DM7, T2, T3, T4, T5, T6, T7, U3, U4, U6, C1, C2, C3, C4, C5, C6, C7, C8, C9, R1, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, P1)/46$
24. $P(K) = (O1, O3, O6, O7, O8, E2, E3, E4, E5, M1, M2, M3, M4, M5, M6, M7, DM1, DM2, DM4, DM5, DM6, DM7, T1, RC10, RC11, U3, U5, C1, C2, C3, C4, C6, C7, R8, R9, R10, R11, R12, P1, P2, P3, P4)/42$
25. $P(L) = (E5, M2, M3, M4, M5, M6, M7)/7$
26. $P(M) = (\sum_{n=1}^8 Q_n)/81$ dimana Q_n adalah pertanyaan metrik dari (O1...O8, E1...E7, M1...M7, DM1...DM7, T1...T9, RC1...RC11, U1...U6, C1...C9, R1...R12, P1...P5)
27. $P(N) = [\sum_{n=1}^4 w_n P(A_n)]$ dimana $w_4 = 0.125, w_5 = 0.125, w_6 = 0.125, w_7 = 0.125, w_8 = 0.125, w_9 = 0.125, w_{10} = 0.04, w_{11} = 0.04, w_{12} = 0.04, w_{13} = 0.125$. Berdasarkan tabel 6-2, bobot 0.04 diperuntukan untuk nilai yang rendah dan 0.125 untuk nilai tinggi
28. $P(O) = [\sum_{n=1}^4 w_n P(A_n)]$ dimana $w_4 = 0.045, w_5 = 0.045, w_6 = 0.045, w_7 = 0.045, w_8 = 0.14, w_9 = 0.14, w_{10} = 0.14, w_{11} = 0.14, w_{12} = 0.14, w_{13} = 0.14$. Berdasarkan tabel 6-2, bobot 0.044 diperuntukan untuk nilai yang rendah dan 0.14 untuk nilai tinggi.

2.4 Flow Map

Flowmap adalah campuran peta dan flow chart yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain. Seperti jumlah orang migrasi, jumlah barang yang diperdagangkan atau jumlah paket dalam jaringan. *Flowmap* membantu analisis dan programmer untuk memecahkan masalah kedalam segmen atau bagian yang lebih kecil dan membantu dalam menganalisis alternatif-alternatif dalam pengoperasian.

2.5 Use Case

Use case diagram atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan system informasi yang akan di buat. Singkatnya, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam system informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

2.6 ERD

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity-Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Entity-Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Untuk menggambarkan ERD digunakan beberapa notasi dan simbol. Tabel 2-3 menjelaskan tentang simbol-simbol dalam ERD[1].

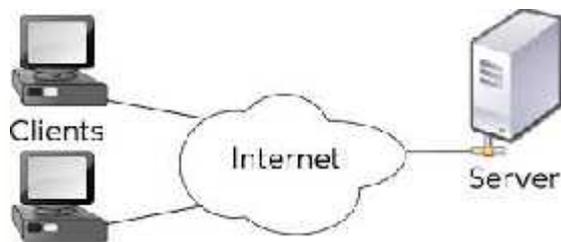
2.7 Aplikasi Web

Aplikasi web adalah suatu aplikasi yang diakses menggunakan *web browser* melalui suatu jaringan seperti internet atau intranet. Jadi, ada tiga komponen untuk menjalankan aplikasi web, yaitu *web client*, *web server*, dan jaringan.

Aplikasi web sangat populer karena kemudahan tersedianya aplikasi client untuk mengaksesnya tanpa tergantung pada platform apapun. Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi aplikasi pada client yang jumlahnya mungkin mencapai ribuan merupakan alasan kunci popularitasnya. Dimana ada web browser maka disana kita bisa mengakses aplikasi web tersebut. Aplikasi web dapat dibangun dengan berbagai bahasa pemrograman, diantaranya PHP, ASP, dan tentu saja JAVA.

Aplikasi web merupakan aplikasi yang menggunakan arsitektur *client-server*. Pada jenis arsitektur ini, sebuah program client (*web browser*) terhubung pada sebuah server untuk bisa mengakses *resource*/sumber daya yang disediakan oleh server. Aplikasi web adalah aplikasi yang sifatnya menunggu. Aplikasi web akan melakukan proses

jika menerima *request*, kemudian akan mengirimkan hasilnya yang disebut *response*. *Request* adalah informasi yang dikirimkan oleh client ke server. *Response* adalah informasi yang dikirimkan server ke *client*[2].



Gambar 2-2 Arsitektur client-server pada aplikasi web[2].

2.8 HTML

HTML kependekan dari **H**yper **T**ext **M**arkup **L**anguage. Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam *web browser*. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet[6].

HTML (Hyper Markup Language) yaitu sebuah bahasa scripting yang dapat menghasilkan halaman website pada dasarnya HTML merupakan bahasa script dasar yang berjalan bersama berbagai bahasa pemrograman lainnya. Kode HTML tidak dapat dijadikan sebagai file executable program, karena HTML hanyalah sebuah bahasa Scripting yang dapat berjalan apabila di jalankan didalam browser seperti Internet Explorer, Netscape Navigator, Opera, Mozilla dan lain-lain[7].

2.9 PHP

Php adalah sebuah bahasa pemrograman yang berjalan dalam sebuah webserver, diciptakan oleh programmer unix dan perl pada bulan Agustus – September 1994 oleh Rasmus Lerdorf. PHP dapat melakukan apa saja yang bisa dilakukan oleh pemrograman CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan data cookies. PHP juga dapat berjalan pada semua jenis system informasi operasi antara lain Linux dan Microsoft, serta dapat berjalan pada beberapa webserver antara lain Apache, IIS, Personal Web Server, Caudium[8].

PHP (Hypertext preprocessor) adalah sebuah bahasa pemrograman yang bekerja dalam sebuah webserver. Dimana script-script *PHP* harus tersimpan dalam sebuah *server* dan dieksekusi atau di proses dalam *server* tersebut, dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis.

Melihat perkembangan teknologi *website*, PHP digunakan oleh *developer website* hampir diseluruh dunia karena bahasa pemrograman PHP bersifat *Open Source* serta lebih dinamis dan mampu berjalan pada sistem operasi apapun. Hal inilah yang menyebabkan bahasa pemrograman PHP berkembang dengan pesat. Jadi dapat disimpulkan bahwa PHP merupakan bahasa pemrograman *website* yang dapat digunakan untuk desain sebuah web menjadi interaktif dan dinamis serta mampu dijalankan dengan kecepatan tinggi dalam sebuah server[9].

2.10 Code Igniter(CI)

CodeIgniter adalah *framework* pengembangan aplikasi (*application Development Framework*) dengan menggunakan PHP, suatu kerangka pembuatan program dengan menggunakan PHP. Pengembang dapat langsung menghasilkan program dengan cepat, dengan mengikuti kerangka kerja untuk membuat yang telah diaplikasikan oleh framework CI ini. Dengan adanya kerangka untuk bekerja atau membuat program maka seorang pemrogram tidak perlu membuat program dari awal, karena CI menyediakan sekumpulan library yang banyak diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses librarinya. Pemrograman dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan. Beberapa fitur yang ada dalam CodeIgniter[9]:

- a. Sistem berbasis Model-View-Controller.
- b. Benar-benar framework yang ringan.
- c. Memiliki fitur class database yang mendukung beberapa platform.
- d. Dukungan database dengan active record.
- e. Form dan validasi data.
- f. Keamanan dan filtering XSS.
- g. Manajemen sesi (session management).

2.11 Black Box Testing

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk

menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak yaitu unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga. Metode ujicoba blackbox memfokuskan pada keperluan fungsional dari software. Karena itu ujicoba blackbox memungkinkan pengembangan software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba blackbox bukan merupakan alternatif dari ujicoba whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode whitebox. Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya[10]:

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan performa.
5. Kesalahan inialisasi dan terminasi.

2.12 User Acceptance Test

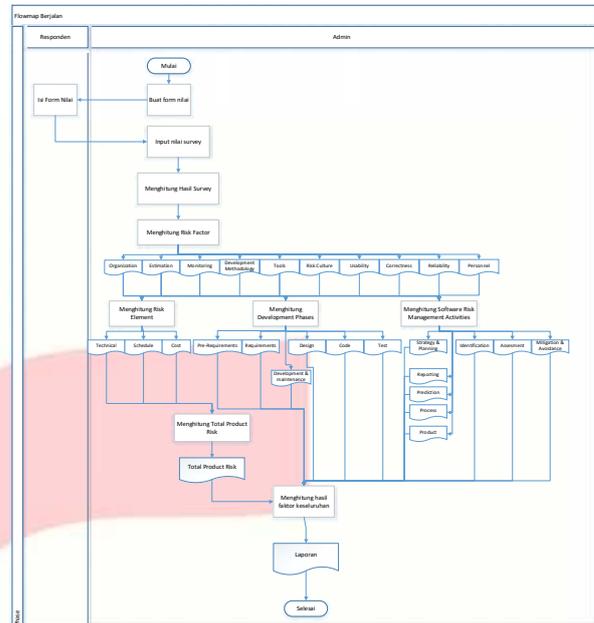
User Acceptance Test (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa software yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (testing) sudah bisa dianggap memenuhi kebutuhan dari pengguna.[13] Proses UAT didasarkan pada dokumen requirement yang disepakati bersama. Dokumen requirement adalah dokumen yang berisi lingkup pekerjaan software yang harus dikembangkan, dengan demikian maka dokumen ini semestinya menjadi acuan untuk pengujian[11].

3. Analisis dan Perancangan

3.1. Flowmap Sistem Berjalan

Sistem yang berjalan saat ini untuk penilaian resiko pengembangan perangkat lunak memiliki dua user yaitu koordinator IT dan responden. Koordinator IT berperan untuk mengelola sistem. Untuk pengisian kuisioner dan perhitungan hasil kuisioner masih manual dengan menggunakan *microsoft excel* sehingga masih kurang efektif karena tidak ada tempat penyimpanan data untuk hasil perhitungan seperti database. Perhitungan dimulai dari perhitungan *risk factor*, *risk element*, *development*

phases, dan *software risk management activities*. Setelah semua perhitungan selesai, akan menghasilkan *total product risk*. Setelah di dapatkan semua nilai dari setiap faktor, maka hasil akhirnya adalah sebuah laporan yang berisi hasil perhitungan.



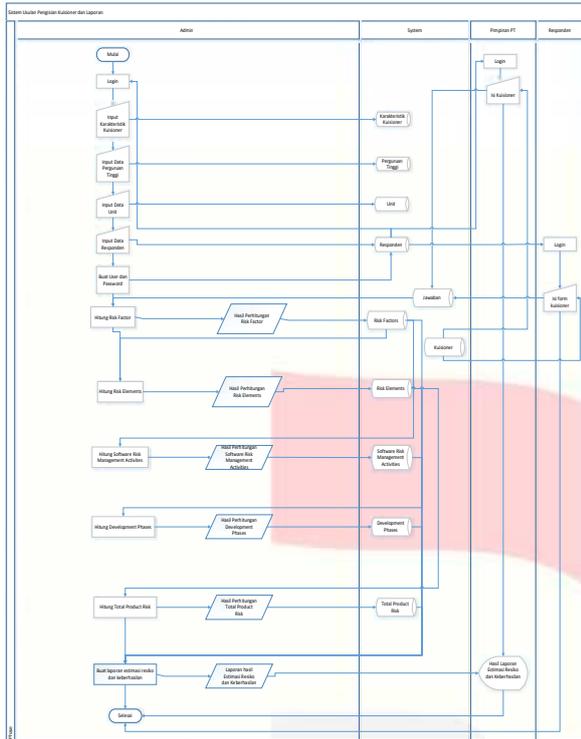
Gambar 3-1 Flowmap berjalan

3.2 Analisis Kebutuhan Sistem(atau Produk)

3.2.1 Flowmap usulan pengisian kuisioner dan laporan

Aplikasi penilaian resiko teknologi informasi memiliki Tiga user dengan kebutuhan yang berbeda-beda. Tiga user ini adalah responden, koordinator IT, pimpinan perguruan tinggi yang mengelola sistem atau yang berperan dalam sistem. Setelah responden mengisi kuisioner perhitungan akan dilakukan dimulai dari *risk factor* dan menghasilkan sepuluh unsur yaitu *organization*, *estimation*, *monitoring*, *development methodology*, *tools*, *risk culture*, *usability*, *correctness*, *reliability*, dan *personel*. Kemudian diikuti dengan perhitungan *risk element*, *development phases*, dan *software risk management activities* dan *total product risk*. *Risk element* menghasilkan tiga unsur yang nantinya akan di gunakan untuk menghitung total product risk, tiga unsur itu meliputi *technical*, *cost*, dan *schedule*. *Development phases* menghasilkan enam unsur yaitu *pre-requirements*, *requirements*, *design*, *code*, *test*, dan *development & maintenance*. *Software risk management risk activities* menghasilkan delapan unsur yaitu *identification*, *strategy & planning*, *assesment*, *mitigation & avoidance*, *reporting*, *prediction*, *process*, dan *product*. Perhitungan *development phases* dan *software risk management activities* mengacu pada hasil perhitungan dari unsur *risk factor*. Setelah di dapatkan nilai dari semua unsur maka di buatlah laporan berdasarkan peluang

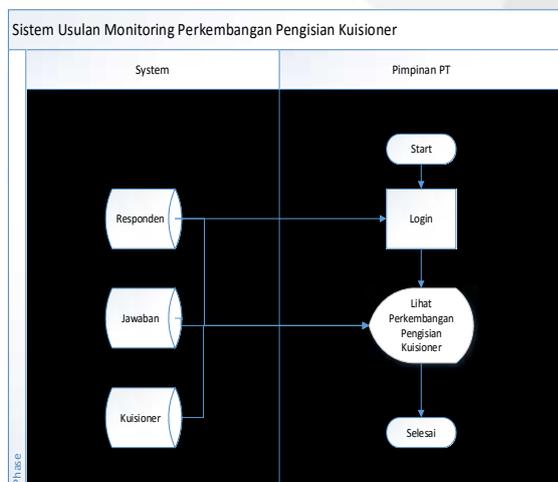
terkecil dari setiap unsur. Gambar 3-2 menggambarkan sistem pada aplikasi penilaian resiko teknologi informasi.



Gambar 3-2 Flowmap usulan pengisian kuisioner

3.2.1 Flowmap usulan monitoring pengembangan kuisioner

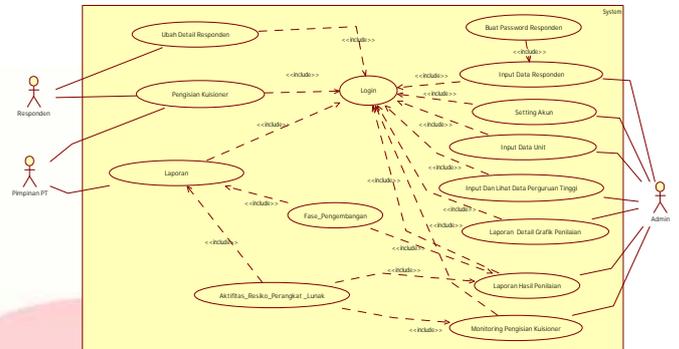
Pimpinan perguruan tinggi memiliki hak akses untuk melihat perkembangan pengisian kuisioner. Hak akses yang dimiliki yaitu dapat melihat berapa persen responden yang sudah mengisi kuisioner dan responden mana yang belum mengisi kuisioner. Serta berapa kuisioner yang telah diisi maupun belum diisi oleh responden.



Gambar 3-3 Flowmap usulan monitoring perkembangan pengisian kuisioner

3.2.3. Use Case Diagram

Berikut merupakan Use Case dari Aplikasi Penilaian Resiko Teknologi Informasi di Perguruan Tinggi Berdasarkan *Software Engineering Risk Management*.



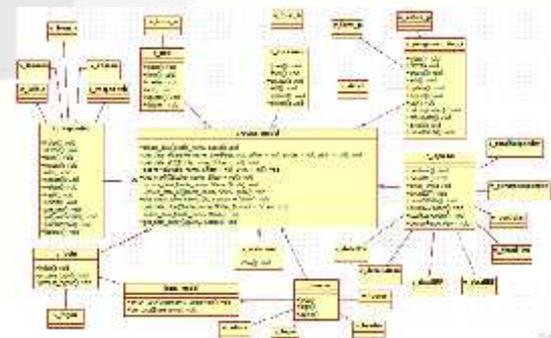
Gambar 3-3 Use Case

Pada pengerjaan aplikasi ini, dari memasukkan data perguruan tinggi, data unit, data responden, ubah profil responden, dan menampilkan kuisioner untuk responden dilakukan secara bersama-sama. Sedangkan untuk pengerjaan perhitungan setiap elemen dibagi menjadi 2 modul yaitu modul A dan Modul B.

Secara keseluruhan perhitungan SERIM dibagi menjadi lima elemen (Risk Factor, Risk Element, Total Product Risk, Development Phases, Software Risk Management Activities). Pada modul ini penulis akan mengerjakan dua elemen yaitu Development Phases, Software Risk Management Activities dan laporan. Dimana perhitungan dua elemen ini didapat dari nilai pertanyaan yang ada pada kuisioner (Risk Factor). Laporan berisi penjelasan mengenai nilai peluang terkecil dan rekomendasi.

3.2.4 Diagram Kelas

Perancangan diagram kelas secara umum yang



digunakan pada aplikasi penilaian pembangunan perangkat lunak di perguruan tinggi berbasis web berdasarkan *software engineering risk management*, adalah sebagai berikut:

Gambar 3-6 Diagram Kelas

3.2.7 Skema Relasi



Gambar 4-3 Beranda Responden
Beranda Pimpinan PT

Pada tampilan pimpinan perguruan tinggi terdapat menu yang sama dengan tampilan responden. Yang membedakan adalah pada tampilan pimpinan perguruan tinggi terdapat menu laporan yang memiliki sub menu laporan penilaian dan laporan persen responden.

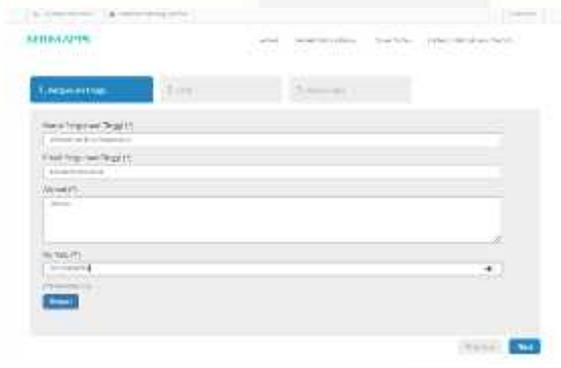


Gambar 4-4 Beranda Pimpinan PT

Menu Pada Beranda Admin

Olah data:

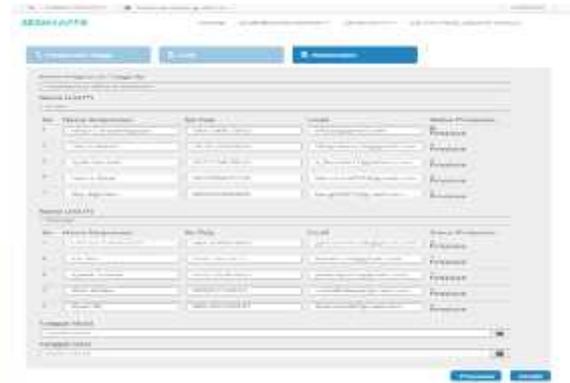
1. Data Perguruan Tinggi



Gambar 4-5 Input Data Perguruan Tinggi
Pada tampilan ini data yang dimasukkan untuk perguruan tinggi yaitu nama perguruan tinggi, email perguruan tinggi, dan nomor telepon perguruan tinggi.



Gambar 4-6 Input data unit
Pada tampilan ini data yang dimasukkan adalah nama unit dan jumlah responden.



Gambar 4-7 Input data responden
Pada tampilan ini data yang dimasukkan adalah data responden, tanggal mulai, dan tanggal akhir. Tanggal mulai dan tanggal akhir adalah waktu dimana dimulainya pengisian kuisioner oleh responden.

2. Kelola Responden



Gambar 4-8 Kelola Responden
Pada tampilan ini admin dapat melihat daftar responden dan dapat mengetahui responden yang menjadi pimpinan perguruan tinggi.

Dashboard Admin:

1. Laporan Total Responden



Gambar 4-8 Laporan total responden
Pada tampilan ini, admin dapat melihat berapa banyak responden yang telah mengisi kuisioner.

2. Laporan Detail Kuisioner



Gambar 4-9 Laporan detail kuisisioner
 Pada tampilan ini, admin dapat melihat jumlah responden yang telah mengisi kuisisioner berdasarkan faktor resiko yang telah dikelompokkan.

3. Laporan Penilaian

Pada tampilan ini, admin dapat melihat hasil penilaian seluruh element. Seluruh element meliputi *risk factor, risk element, total product risk, development phases, software risk management activities.*



Gambar 4-10 Laporan Penilaian

4. Detail Laporan Penilaian

Tampilan ini menampilkan tabel nilai dari setiap element. Dimana hasil didapatkan dari perhitungan jawaban dan bobot yang telah ditentukan.

Gambar 4-11 Detail laporan penilaian

5. Detail Perguruan Tinggi

Pada menu ini admin dapat melihat detail perguruan tinggi dari setiap unit. Detail meliputi data responden yang dapat dicetak per unit.

No	Nama Universitas	No Telp	Email	No Responden
1	UIN Sunan Kalijaga	021-3610100	info@uinsu.ac.id	10 Responden
2	UIN Sunan Gunung Djati	021-2534300	info@uinsgd.ac.id	10 Responden
3	UIN Sunan Drajat	031-8321111	info@uinsd.ac.id	10 Responden

Gambar 4-12 Detail Perguruan Tinggi



Gambar 4-14 Cetak detail perguruan tinggi

Menu Pada Beranda Responden

1. Profil Responden

Pada menu ini responden dapat melakukan ubah profil seperti ubah password, nama, email, dan nomor telepon. Selain itu responden dapat melihat kapan terakhir responden melakukan ubah password.



Gambar 4-15 profil responden

2. Isi Kuisisioner

Pada menu ini responden mengisi kuisisioner. Pada menu ini responden mengisi kuisisioner. Pengisian kuisisioner dapat dilakukan secara bertahap dengan jangka waktu tujuh hari. Pada tampilan kuisisioner terdapat pengingat waktu pengisian kuisisioner. Pengisian nilai kuisisioner adalah 0.0, 0.5, 0.1. Isi dari kuisisioner berupa pertanyaan yang berkaitan dengan organization, estimation, monitoring, development methodology, tools, risk culture, usability, correctness, reliability, dan personel.



Gambar 4-16 Isi kuisisioner

Menu Pada Pimpinan Perguruan Tinggi

Menu Pimpinan:

1. Profile

Pada menu ini pimpinan perguruan tinggi dapat melakukan ubah profil seperti ubah kata sandi, nama, email, dan nomor telepon. Selain itu pimpinan perguruan tinggi dapat melihat kapan terakhir pimpinan melakukan ubah kata sandi.



Gambar 4-17 Profile pimpinan

2. Isi Kuisisioner

Pada menu ini pimpinan perguruan tinggi mengisi kuisisioner. Pengisian kuisisioner dapat dilakukan secara bertahap dengan jangka waktu tujuh hari. Pada tampilan kuisisioner terdapat pengingat waktu pengisian kuisisioner. Pengisian nilai kuisisioner adalah 0.0, 0.5, 0.1. Isi dari kuisisioner berupa pertanyaan yang berkaitan dengan organization, estimation, monitoring, development methodology, tools, risk culture, usability, correctness, reliability, dan personel.



Gambar 4-18 Isi Kuisisioner pimpinan

Laporan:

1. Laporan Penilaian

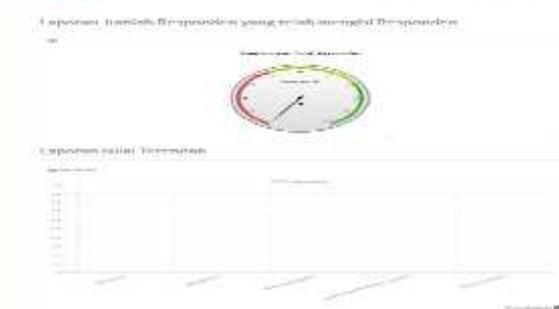
Pada menu ini pimpinan dapat melihat hasil dari perhitungan yang dilakukan oleh sistem. Laporan berupa grafik dan detail penilaian.



Gambar 4-19 Laporan penilaian pimpinan

2. Laporan Persen Responden

Pada menu ini, pimpinan perguruan tinggi dapat melihat persentase responden yang telah mengisi kuisisioner. Semakin tinggi persentase, maka semakin banyak responden yang telah mengisi kuisisioner.



Gambar 4-20 Laporan persen responden

5. Penutup

5.1 Kesimpulan

Setelah melakukan kegiatan analisis kebutuhan, desain, perancangan kode program, dan pengujian terhadap aplikasi penilaian pembangunan perangkat lunak di perguruan tinggi berdasarkan software engineering risk management penulis mengambil kesimpulan diantaranya adalah:

1. Aplikasi ini mampu memperkirakan resiko yang akan muncul selama pengerjaan proyek.
2. Aplikasi ini memudahkan perguruan tinggi dalam penyebaran kuisisioner ke responden.
3. Aplikasi ini memudahkan responden dalam pengisian kuisisioner.
4. Aplikasi ini mampu menghasilkan 28 keluaran nilai dari 5 element utama secara detail.
5. Aplikasi ini menghasilkan laporan akhir yang berisikan deskripsi dari setiap resiko.

5.2 Saran

Berdasarkan hasil pembangunan proyek akhir ini, penulis menyampaikan beberapa saran untuk pengembangan selanjutnya yaitu:

- 1 Aplikasi ini diharapkan dapat digunakan oleh semua institusi, tidak hanya perguruan tinggi.
- 2 Aplikasi ini diharapkan dapat digunakan pada smartphone.

[13] Raharjo. 2001. Belajar Pemrograman Web. Bandung: Modula.

[14] Arief Ramadhan, S.Kom. 2006. Pemrograman Web Database dengan PHP dan MySQL. Jakarta: PT Elex Media Komputindo.

Daftar Pustaka:

- [1] Rosa A.S., M. Salahuddin. 2014. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.
- [2] Luwis. (2002) Pemrograman Web Aplikatif dengan Java. Jakarta: Elex Media Komputindo
- [3] Rot, Artur. (2009) Enterprise Information Technology Security: Risk Management Perspective, Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II WCECS 2009, October 20-22, 2009, San Francisco, USA
- [4] Ahlan, Abdul Rahman., Arsyad, Yusri. (2012) Information Technology Risk Management: The case of the International Islamic University Malaysia, Journal Of Research And Innovation In Information Systems.
- [5] Karolak, Dale Walter. 1996. Software Engineering Risk Management. Los Alamitos: IEEE Computer Society Press.
- [6] Nugroho, bunafit. (2009). Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL. Yogyakarta : Gava Media.
- [7] Riyanto. (2011). Sistem Informasi Penjualan dengan PHP dan MySQL. Yogyakarta : Gava Media.
- [8] H, Rafiza. 2006. Panduan dan Refrensi Kamus Fungsi PHP 5. Jakarta: PT Elex Media Komputindo.
- [9] Madcoms. Menguasai HTML, CSS, PHP dan MySQL Indonesia. 2011.
- [10] MT A.Suhendar.S.Si dan Hariman Gunadi.S.Si., Visual Modeling Menggunakan Rational Rose. Bandung: Informatika, 2001.
- [11] Wanhen., 2010., Apakah Yang Dimaksud Dengan User Acceptance Test?. <http://www.exforsys.com/tutorials/testing/what-is-user-acceptance-testing.html>. Diakses pada tanggal 13 Maret 2016
- [12] Jurnal Teknologi Industri Vol. XI No. 2 April 2007:13-24