

**PENGUJIAN PERANGKAT LUNAK DENGAN
TECHNICAL REVIEW**



Disusun Oleh:

N a m a : Arya Hafiz Saputra

NIM : 13523212

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING
PENGUJIAN PERANGKAT LUNAK DENGAN
TECHNICAL REVIEW



الجامعة الإسلامية
Yogyakarta, 05 Februari 2018
Pembimbing.

(Hansari Prihantoro Putro, S.I., M.T.)

HALAMAN PENGESAHAN DOSEN PENGIJI

**PENGUJIAN PERANGKAT LUNAK DENGAN
TECHNICAL REVIEW**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta, 5 Januari 2018

Tim Penguji

Hanson Prihantoro Putro, S.T., M.T.

Anggota 1

Almed Hamzah, S.T., M.Eng.

Anggota 2

Ilari Sctiaji, S.Kom., M.Eng.

Mengetujui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Indonesia



Hendrik, S.T., M.Eng.

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Arya Hafiz Saputra

NIM : 13523212

Tugas akhir dengan judul:

**PENGUJIAN PERANGKAT LUNAK DENGAN
TECHNICAL REVIEW**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 05 Februari 2018



(Arya Hafiz Saputra)

HALAMAN PERSEMBAHAN

Alhamdulillah Robbil 'Alamin puji syukur atas segala nikmat dan karunia yang Allah SWT berikan kepada saya sehingga tugas akhir ini dapat terselesaikan dengan baik. Atas semua dukungan dan bantuan yang telah diberikan, saya persembahkan tugas akhir ini untuk,

Kedua orang tua saya yang tercinta, Ir. Haryanto. Dan Ibu Ir. Istiana ,
Adik-adikku, Farhan Arief Saputra, Kayla Salsabila dan
Adzam Nabil Saputra serta Keluarga Besar,

Sahabat-sahabat saya, Tim Delegasi FTI,
Seluruh teman-teman ETERNITY 2013 yang menjadi teman seperjuangan di bangku
perkuliahan,

Dan seluruh teman-teman yang tidak bisa disebutkan satu per satu.

HALAMAN MOTO

“Maka sesungguhnya beserta kesulitan itu ada kemudahan. Sesungguhnya beserta kesulitan itu ada kemudahan.”

(QS. Al-Insyirah : 5-6)

“Allah akan meninggikan orang-orang yang beriman di antara dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(QS. Al-Mujadalah : 11)

“Mencari Ilmu itu adalah wajib bagi setiap muslim laki-laki maupun muslim perempuan.”

(HR. Ibnu Abdil Barr)

“Barang siapa yang keluar untuk mencari ilmu maka ia berada di jalan Allah hingga ia pulang.”

(HR. Tumudzi)

“Barang siapa yang menempuh jalan untuk mencari suatu ilmu, Niscaya Allah Memudahkannya ke jalan menuju surga.”

(HR. Muslim)

“Menuntut ilmu itu diwajibkan bagi setiap orang islam.”

(HR. Ibnu majah, Al-Baihaqi, Ibnu Abdil Barr, dan Ibnu Adi, dari Anas bin malik)

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, penulis ucapkan puji syukur kepada Allah SWT yang telah memberikan rahmat, hidayah, serta karunia-Nya, sehingga laporan Tugas Akhir dapat penulis selesaikan. Tidak lupa shalawat serta salam kami ucapkan kepada junjungan nabi Allah Nabi Muhammad SAW, beserta para keluarga dan sahabatnya yang telah membawa kita dari zaman kegelapan menuju zaman terang benderang.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Jurusan Teknik Informatika Universitas Islam Indonesia. Adapun Tugas Akhir kami mengenai “Pengujian Perangkat Lunak Dengan Technical Review.”

Pelaksanaan Tugas Akhir ini merupakan salah satu mata kuliah wajib dari jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia dan juga merupakan sarana bagi penulis untuk menambah wawasan serta pengalaman dalam menerapkan keilmuan, sesuai dengan yang ditambah di bangku perkuliahan.

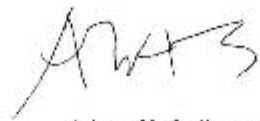
Oleh, Karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Orang tua dan keluarga penulis atas segala doa dan dukungan selama penulis melakuakn Tugas Akhir.
2. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Hanson Prihantoro Putro, S.T., M.T., selaku Dosen Pembimbing 1 Tugas Akhir di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Segenap keluarga besar teman-teman di Fakultas Teknologi Industri terutama dari Jurusan Teknik Informatika Universitas Islam Indonesia yang telah memberikan bantuan dan dukungannya.
5. Semua pihak yang telah banyak membantu kami dalam pelaksanaan Tugas Akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaaan di lapangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikumun Warahmatullahi Wabarakatuh

Yogyakarta, 5 Januari 2018



(Arya Hafiz Saputra)

SARI

Pembuatan aplikasi selalu dilakukan dengan menggunakan banyak tahapan. Tahapan-tahapan utama dari pembuatan aplikasi secara sederhana adalah pengumpulan data, pembuatan desain, implementasi, dan pengujian. Dalam tahap pengujian biasanya pembuat aplikasi hanya menggunakan teknik-teknik seperti *white box* maupun *black box* yang dilakukan ketika aplikasi sudah dalam tahap hampir jadi. Padahal pengujian dapat dilakukan pada tahap-tahap awal yaitu dari tahap pembuatan desain dengan menggunakan teknik yang dinamakan *technical review*. *Technical review* merupakan suatu pengujian yang dilakukan dengan cara melakukan pertemuan antara pembuat aplikasi dan peninjau di sebuah ruangan rapat dan membahas beberapa bagian-bagian penting dari aplikasi untuk ditemukan permasalahan sebelum aplikasi diimplementasikan. *Technical review* sangat mudah dilakukan namun sering dilupakan dalam proses pembuatan aplikasi.

Penelitian ini dilakukan dengan menggunakan tinjauan terhadap aplikasi mahasiswa lain yang dibuat sebagai Tugas Akhir di Jurusan Teknik Informatika. Tinjauan dilakukan mulai dari tahap pembuatan desain aplikasi hingga tahap akhir pembuatan aplikasi. Tinjauan dilakukan di Perpustakaan Terpadu Universitas Islam Indonesia yang terletak di Jalan Kaliurang Km 14,5 Sleman DIY. Setiap tinjauan yang dilakukan menghasilkan laporan berbentuk formulir yang berisi hal yang ditinjau, siapa yang melakukan peninjauan, dan kesalahan-kesalahan apa saja yang ditemukan disetiap pertemuan yang dilakukan. Setelah setiap tahap pembuatan aplikasi selesai ditinjau akan dibuat sebuah *traceability matriks* yang digunakan untuk mengecek konsistensi dari pembuatan aplikasi dimulai dari 1)*use case*, 2)*activity diagram*, 3)*desain database*, 4)*desain interface*, hingga 5)*source code*. Setelah *traceability matriks* selesai dibuat, peninjau dapat mengetahui aplikasi yang dibuat sudah sesuai dengan rancangan yang sudah ditetapkan atau tidak. Setiap peninjauan memiliki metrik-metrik penilaian yang berupa durasi pertemuan dilakukan dan kerapatan kesalahan. Durasi pertemuan akan semakin baik jika lebih sedikit dan kerapatan masalah akan lebih baik jika hasilnya lebih kecil juga. *Technical review* yang baik dapat menghasilkan waktu pembuatan aplikasi yang lebih sedikit dan juga mengurangi munculnya error ketika aplikasi sudah selesai dibuat.

Kata kunci: *Technical review, traceability matriks, error density, software development, software testing*

GLOSARIUM

| | |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Technical Review | Proses dilakukanya pengecekan kesalahan di dalam bagian tertentu dalam proses pembuatan aplikasi. Pengecekan dilakukan saat pertemuan antara pembuat aplikasi dan peninjau dalam satu ruangan. |
| Traceability Matriks | Tabel yang dibuat untuk melakukan penelusuran setiap tahapan pembuatan aplikasi apakah sudah sesuai satu sama lain atau tidak. |
| Error Density | Merepresentasikan kesalahan-kesalahan yang ditemukan untuk tiap unit kerja produk yang ditinjau. |
| Use Case | Sebuah cara yang menggambarkan kebutuhan fungsional dari sebuah sistem yang akan dibuat. |
| Activity Diagram | Diagram yang menggambarkan aktivitas yang dilakukan di dalam aplikasi. |
| Database | Kumpulan data di dalam komputer yang digunakan untuk menghasilkan informasi. |
| Interface | Tampilan di dalam aplikasi yang langsung dilihat oleh pengguna. |
| User | Pengguna sebuah aplikasi. |

DAFTAR ISI

| | |
|----------------------------------------------------------|-------------------------------------|
| HALAMAN JUDUL | i |
| HALAMAN PENGESAHAN DOSEN PEMBIMBING..... | Error! Bookmark not defined. |
| HALAMAN PENGESAHAN DOSEN PENGUJI | Error! Bookmark not defined. |
| HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR..... | Error! Bookmark not defined. |
| HALAMAN PERSEMBAHAN | iv |
| HALAMAN MOTO | vi |
| KATA PENGANTAR..... | vii |
| SARI | viii |
| GLOSARIUM | x |
| DAFTAR ISI | xi |
| DAFTAR TABEL | xiii |
| DAFTAR GAMBAR..... | xiv |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan Penelitian | 2 |
| 1.5 Manfaat Penelitian | 3 |
| 1.6 Metode Penelitian | 3 |
| 1.7 Sistematika Penulisan | 3 |
| BAB II LANDASAN TEORI | 5 |
| 2.1 Tinjauan Pustaka | 5 |
| 2.2 Pengujian Perangkat Lunak | 6 |
| 2.3 <i>Technical Review</i> | 6 |
| 2.3.1 Metrik-metrik Peninjauan dan Penggunaannya | 7 |
| 2.3.2 Menganalisis Metrik-metrik | 8 |
| 2.3.3 Tinjauan-tinjauan Informal..... | 8 |
| 2.3.4 Tinjauan-tinjauan Teknis Yang Bersifat Formal..... | 9 |
| 2.4 Konsep Management Risiko | 9 |
| BAB III METODOLOGI | 11 |
| 3.1 Prosedur-prosedur Pengambilan Data..... | 11 |

| | | |
|-----------------------------------|------------------------------------------------------------------------------------|----|
| 3.2 | <i>Technical Review</i> | 11 |
| 3.2.1 | Pertemuan-pertemuan Peninjauan | 12 |
| 3.2.2 | Pelaporan Hasil Peninjauan dan Pemeliharaan Catatan-catatan Hasil Peninjauan | 12 |
| 3.2.3 | Panduan-panduan Peninjauan | 13 |
| 3.3 | <i>Technical Review</i> Dalam Fase Pengembangan Perangkat Lunak | 15 |
| 3.3.1 | Analisis Kebutuhan | 15 |
| 3.3.2 | Desain Sistem | 15 |
| 3.3.3 | Desain <i>Database</i> (TB) | 16 |
| 3.3.4 | Desain <i>Interface</i> (UI) | 17 |
| BAB IV HASIL DAN PEMBAHASAN | | 18 |
| 4.1 | Persiapan <i>Technical Review</i> | 18 |
| 4.1.1 | Pertemuan-pertemuan Peninjauan | 18 |
| 4.1.2 | Pelaporan Hasil Peninjauan | 18 |
| 4.1.3 | Panduan-Panduan Peninjauan | 19 |
| 4.2 | Laporan Pertemuan <i>Technical Review</i> | 20 |
| 4.3 | Hasil <i>Technical Review</i> | 21 |
| 4.3.1 | Analisis Kebutuhan | 22 |
| 4.3.2 | <i>Use Case Diagram</i> | 23 |
| 4.3.3 | <i>Activity Diagram</i> | 24 |
| 4.3.4 | Desain <i>Database</i> | 26 |
| 4.3.5 | Desain <i>Interface</i> | 27 |
| 4.3.6 | <i>Source Code</i> | 29 |
| 4.3.7 | <i>Review</i> Pengujian | 30 |
| 4.4 | Evaluasi Hasil <i>Technical Review</i> | 33 |
| BAB V KESIMPULAN DAN SARAN | | 35 |
| 5.1 | Kesimpulan | 35 |
| 5.2 | Saran | 35 |
| DAFTAR PUSTAKA | | 36 |
| LAMPIRAN | | 38 |

DAFTAR TABEL

| | |
|--------------------------------------------------------------------------|----|
| Tabel 3.1 Prosedur Pengambilan Data..... | 11 |
| Tabel 4.1 Hasil Pertemuan <i>Technical Review</i> Berdasarkan Waktu..... | 20 |
| Tabel 4.2 Hasil Pertemuan <i>Technical Review</i> Berdasarkan Topik..... | 21 |
| Tabel 4.3 Hasil <i>Review Use Case Diagram</i> | 23 |
| Tabel 4.4 Hasil <i>Review Activity Diagram</i> | 25 |
| Tabel 4.5 Hasil <i>Review Desain Database</i> | 26 |
| Tabel 4.6 Hasil <i>Review Desain Interface</i> | 28 |
| Tabel 4.7 Hasil <i>Review Source Code</i> | 29 |
| Tabel 4.8 <i>Test Case</i> Aplikasi..... | 30 |
| Tabel 4.9 Hasil <i>Traceability</i> Matriks..... | 32 |

DAFTAR GAMBAR

| | |
|-------------------------------------------------------|----|
| Gambar 2.1 Proses Manajemen Risiko (NIST, 2012) | 10 |
| Gambar 3.1 Proses <i>Technical Review</i> | 14 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perangkat lunak atau *software* adalah produk yang dibuat oleh profesional dalam jangka panjang. *Software* meliputi program yang dieksekusi di dalam komputer dengan ukuran dan bentuk yang berbeda, isi yang direpresentasikan sebagai eksekusi program komputer, dan mendeskripsikan informasi baik fisik maupun visual yang meliputi virtualisasi dari media elektronik (Pressman, 2010). *Software* juga dapat disebut sebagai instruksi yang ketika dieksekusi akan menyediakan fitur, fungsi, dan performa yang diinginkan (Pressman, 2010). Aplikasi yang digunakan pastinya akan mengalami proses pengembangan aplikasi. Proses inti dari pengembangan aplikasi terdiri dari 4, yaitu analisis, desain, implementasi, dan pengujian. Analisis merupakan penguraian dari suatu sistem yang dilakukan untuk mengidentifikasi masalah dan solusinya. Desain adalah tahapan setelah analisis untuk mendefinisikan kebutuhan-kebutuhan fungsional sistem. Implementasi merupakan proses pembuatan sistem setelah proses analisis dan desain sudah selesai. Pengujian merupakan tahap dilakukannya pemeriksaan terhadap aplikasi, apakah terdapat kesalahan maupun kekurangan dari sebuah aplikasi.

Proses pengembangan aplikasi di atas ada kalanya tidak dilaksanakan dengan baik, yaitu dilewatkannya pengujian *technical review*. *Technical review* dilakukan dengan cara mencari dan membenarkan kesalahan yang ada sebelum *software* diberikan kepada pelanggan. *Technical review* dibutuhkan seperti halnya pensil membutuhkan penghapus, yaitu untuk membenarkan kesalahan yang dibuat. Alasan lainnya *technical review* dibutuhkan adalah walaupun beberapa orang dapat mengetahui *error* kecil yang mereka buat sendiri, *error* yang besar dapat saja dilewatkan oleh sang pembuat namun dapat ditangkap oleh orang lain (Pressman, 2010). *Technical review* kurang populer dibandingkan dengan metode pengujian lainnya seperti *black box* maupun *white box*. Hal ini disebabkan karena *technical review* dilakukan dengan melakukan pertemuan antara peninjau dan pembuat aplikasi sedangkan metode lainnya tidak melakukan pertemuan. Kebanyakan pembuat aplikasi lebih memilih menggunakan metode lain dibandingkan *technical review* karena merasa melakukan pertemuan dengan peninjau tidak memberikan dampak terhadap proses pembuatan aplikasi. Padahal dengan dilakukannya *technical review* ini dapat memberikan manfaat yang tidak terduga kepada pihak pengembang perangkat lunak. Kesalahan yang terjadi dapat berkurang dengan penerapan *technical review*

yang benar sehingga biaya pembuatan perangkat lunak yang digunakan untuk perbaikan kesalahan juga dapat berkurang.

Beberapa kasus kurang tepatnya *technical review* muncul di Indonesia, contohnya aplikasi Go-Jek. Aplikasi Go-Jek merupakan aplikasi yang digunakan untuk memesan ojek secara online. Go-Jek menambahkan fitur baru untuk aplikasi, namun terjadi permasalahan saat meluncurkan fitur baru tersebut. Hal ini terbukti dengan munculnya berita hilangnya aplikasi Go-Jek di Appstore yang pernah muncul pada bulan April 2017 (Widiartanto, 2017). Dengan penggunaan *technical review* yang baik, seharusnya fitur yang akan ditambahkan tersebut tidak menimbulkan masalah yang harus membuat aplikasi ditarik untuk beberapa hari.

Faktor-faktor di atas mengharuskan para pembuat aplikasi untuk mempertimbangkan teknik pengujian *technical review* seperti apa yang tepat untuk aplikasi mereka. Di dalam penelitian ini akan dilakukan teknik pengujian *technical review* terhadap pengembangan aplikasi dengan tema “Sistem Penilaian Risiko (Ikhsani, 2017).” Dipilihnya aplikasi tersebut karena aplikasi masih di dalam tahap pengembangan sehingga *technical review* dapat diterapkan secara maksimal terhadap aplikasi tersebut. Harapannya dengan dilakukannya *technical review* ini dapat membuat aplikasi yang dibuat akan semakin baik dan kemungkinan terjadinya *error* dapat dikurangi.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, ada beberapa rumusan masalah, yaitu:

- A. Bagaimana pengujian *technical review* dilakukan untuk proses pengembangan perangkat lunak?
- B. Bagaimana pengujian *technical review* yang menyeluruh mampu memberikan manfaat untuk pengembangan perangkat lunak?

1.3 Batasan Masalah

Di dalam penelitian ini akan dilakukan suatu batasan pekerjaan yaitu hanya melakukan pengujian terhadap sebuah pengembangan perangkat lunak “Sistem Penilaian Risiko Bidang Keamanan Informasi pada Sistem Informasi Menggunakan *National Institute of Standard and Technology* (NIST)”.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

- a. Membuat dokumen untuk tatacara melakukan pengujian *technical review* yang benar sehingga metode pengujian ini dapat diterapkan dalam setiap proses pengembangan aplikasi.
- b. Memastikan bahwa aplikasi yang sedang dikembangkan dapat dibuat secara efisien dengan diterapkannya pengujian *technical review*.

1.5 Manfaat Penelitian

Penelitian ini akan menghasilkan suatu dokumen yang dapat digunakan sebagai pedoman melakukan *technical review* untuk pengembangan aplikasi lainnya.

1.6 Metode Penelitian

Technical review adalah suatu teknik pengujian yang dilakukan dengan cara menemukan dan membetulkan kesalahan yang ada sebelum aplikasi dikirim ke pelanggan. *Technical review* terhadap pengembangan aplikasi Sistem Penilaian Risiko dilakukan dengan beberapa tahapan.

Langkah pertama adalah perencanaan. Perencanaan merupakan tahap merencanakan bagian apa di dalam aplikasi yang akan ditinjau serta waktu dan tempat pelaksanaan rapat. Langkah kedua adalah persiapan. Di dalam langkah ini akan disiapkan hal-hal yang akan ditinjau seperti *script* program, data program, anggota rapat, dan lain-lain. Langkah ketiga adalah membuat struktur rapat. Struktur rapat yang dibuat memiliki beberapa hal seperti berapa lama rapat akan dilaksanakan dan menentukan moderator yang akan menentukan jalannya rapat. Langkah keempat adalah mencatat *error*. Salah satu *reviewer* bertugas untuk merekam setiap masalah yang ditemukan dan akan dibahas di bagian koreksi. Langkah kelima adalah membuat koreksi. Menentukan apakah aplikasi akan dikoreksi, dibiarkan, atau dibatalkan karena error yang fatal. Langkah terakhir berupa verifikasi koreksi. Program yang sudah ditinjau dan dibetulkan akan dibuat laporannya dan disimpan.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bagian ini berisi tentang latar belakang permasalahan, identifikasi masalah, menentukan batasan masalah yang akan dibahas, tujuan dan manfaat dari penelitian, asumsi metodologi penelitian, dan sistematika penulisan

BAB II LANDASAN TEORI

Pada bagian ini menjelaskan teori-teori yang digunakan di dalam penelitian. Setiap teori yang digunakan di dalam penelitian akan dijabarkan di bagian ini.

BAB III METODOLOGI PENELITIAN

Pada bagian ini menampilkan metode yang akan digunakan untuk mengumpulkan data penelitian. Penelitian ini akan menggunakan formal *technical review* sebagai sebuah cara untuk mengumpulkan data penelitian.

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini menjelaskan hasil-hasil dari *review* yang telah dilakukan dalam proses pengembangan aplikasi. *Review* yang akan ditampilkan berupa *review* dari *use case* diagram, *activity* diagram, desain *database*, desain interface, dan *source code* dari pengembangan aplikasi.

BAB V KESIMPULAN DAN SARAN

Pada bagian ini akan dijabarkan kesimpulan yang didapatkan setelah menyelesaikan penelitian. Selain itu juga diberikan saran yang memberikan poin-poin yang dapat ditingkatkan untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pengujian perangkat lunak banyak digunakan dalam proses pengembangan perangkat lunak. Pengujian harus dilakukan karena akan muncul masalah jika aplikasi yang masih dalam tahap pengembangan tidak dilakukan pengujian. Salah satu teknik pengujian yang masih kurang digunakan secara maksimal adalah *technical review*.

Pengembangan perangkat lunak memiliki banyak metode untuk menguji sudah layak atau tidaknya sebuah perangkat lunak dilepaskan ke pasaran. Salah satu metode yang sering digunakan adalah metode *white box* dan *black box* (Rouf, 2012). Kedua metode ini memiliki cara kerja yang berbeda sehingga hasil kesalahan yang muncul juga akan berbeda satu sama lain. Kedua metode memiliki kelebihan dan kekurangannya masing-masing. *White box* adalah bentuk pengujian dengan cara pengecekan perancangan dan juga program di dalam perangkat lunak sedangkan pengujian *black box* dilakukan dengan hanya mengamati hasil eksekusi program tanpa melihat isi dari program tersebut.

Sebuah sistem informasi penilaian risiko merupakan sebuah sistem yang digunakan untuk menilai seberapa rentan sebuah perangkat lunak terhadap serangan dari pihak luar. Terdapat perangkat lunak yang dapat digunakan untuk melakukan tugas ini, salah satunya adalah Sistem Penilaian Risiko Berbasis NIST 800-30 (Ikhsani, 2017). Sistem ini menggunakan NIST 800-30 sebagai patokan penilaian. Hal ini mempermudah auditor untuk melakukan pendataan terhadap risiko yang ditemukan.

Technical review tidak hanya dapat dilakukan dalam proses pengembangan perangkat lunak, *technical review* dapat juga dilakukan dalam proyek pengerjaan *Fly Over* (A, 2005). Di dalam penelitian ini disebutkan bahwa dalam pembangunan *Fly Over* dapat menggunakan *technical review* yaitu menentukan waktu berapa lama waktu tercepat kemungkinan selesainya proyek hingga waktu paling lama kemungkinan selesainya proyek. Hal ini menambahkan bukti bahwa *technical review* akan memberikan manfaat dalam proses pengembangan perangkat lunak dari segi efektivitas waktu pengerjaan perangkat lunak.

Dari ketiga penelitian di atas telah didapatkan metode pengujian secara *white box* dan *black box*. Selain itu juga telah ada contoh penerapan penilaian risiko dalam sebuah aplikasi. Hal lain yang telah ada adalah penerapan *technical review* dalam pengerjaan sebuah proyek.

Hal yang belum ada di dalam penelitian adalah pedoman untuk melakukan *technical review* terhadap sebuah aplikasi. Hal itu yang akan menjadi fokus di dalam penelitian ini.

2.2 Pengujian Perangkat Lunak

Perangkat lunak diuji untuk menemukan kesalahan yang dibuat secara tidak sengaja saat perangkat lunak tersebut dirancang dan dibangun. Sebuah strategi untuk pengujian perangkat lunak dikembangkan oleh manajer proyek, rekayasawan perangkat lunak, dan spesialis pengujian.

Pengujian sering memerlukan usaha proyek yang lebih dibanding kegiatan rekayasa perangkat lunak yang lain. Jika pengujian ini dilakukan sembarangan, waktu akan terbuang sia-sia, usaha yang sebenarnya tidak perlu telah terlanjur dikeluarkan, dan bahkan lebih buruk, kesalahan tidak terdeteksi. Karena itu, masuk akal untuk memberntuk strategi sistematis untk pengujian perangkat lunak.

Pengujian dimulai “dari yang kecil” dan berlanjut “ke yang besar”. Maksudnya adalah, pengujian awal berfokus pada komponen tunggal atau sekelompok kecil komponen-komponen yang saling terkait dan baru kemudian dilakukan pengujian untuk mengungkap kesalahan dalam data dan logika pemrosesan yang telah tertanam di dalam komponen-komponen tersebut. Setelah komponen diperiksa, komponen-komponen itu haruslah diintegrasikan sampai sebuah sistem yang lengkap terbangun. Pada titik ini, serangkaian pengujian yang bersifat penting dilaksanakan untuk menemukan kesalahan dalam memenuhi persyaratan pelanggan. Saat kesalahan ditemukan, kesalahan tersebut harus didiagnosis dan dikoreksi dengan menggunakan sebuah proses yang disebut pelacakan kesalahan (*debugging*).

Sebuah spesifikasi pengujian mendokumentasikan pendekatan tim perangkat lunak dalam melakukan pengujian caranya dengan mendefinisikan sebuah rencana yang menggambarkan keseluruhan strategi dan prosedur yang mendefinisikan langkah-langkah pengujian secara spesifik dan jenis pengujian yang akan dilakukan.

Dengan meninjau spesifikasi pengujian sebelum melaksanakan pengujian, anda bisa menilai kelengkapan kasus dan tugas pengujian. Perencanaan dan prosedur pengujian yang efektif akan membawa pada konstruksi perangkat lunak yang teratur dan penemuan kesalahan pada setiap tahap dalam proses konstruksi.

2.3 Technical Review

Technical review merupakan “filter” untuk proses-proses perangkat lunak. Yaitu, tinjauan-tinjauan semestinya diterapkan pada berbagai titik pada perjalanan proses rekayasa perangkat lunak dan semestinya juga bertindak sebagai penyingkap kesalahan-kesalahan (*error*)

dan cacat-cacat program (*defect*) yang kemudian dapat dihilangkan. Penggunaan technical review yang tepat dapat mengurangi waktu dan biaya pengerjaan perangkat lunak.

Technical review pada dasarnya memiliki sifat “memurnikan” produk-produk kerja rekayasa perangkat lunak, termasuk di dalamnya memurnikan spesifikasi-spesifikasi kebutuhan dan model-model perancangan, memurnikan kode-kode yang ditulis dalam bahasa-bahasa pemrograman tertentu yang dipilih, dan memurnikan pengujian data.

Suatu tinjauan (tinjauan untuk hal apa saja) adalah suatu cara menggunakan keragaman di dalam kelompok orang-orang untuk:

1. Menekankan pentingnya perbaikan-perbaikan produk yang merupakan hasil dari seseorang atau pentingnya perbaikan-perbaikan produk yang merupakan hasil dari suatu tim.
2. Melakukan konfirmasi apakah bagian-bagian produk dimana perbaikan dilaksanakan dikehendaki atau tidak, dibutuhkan atau tidak.
3. Mencapai pekerjaan teknis yang lebih seragam, atau paling tidak lebih mudah diramalkan kualitas yang kemudian dapat dicapai tanpa tinjauan-tinjauan, dengan tujuan untuk membuat pekerjaan teknis menjadi lebih mudah untuk dikelola.

2.3.1 Metrik-metrik Peninjauan dan Penggunaannya

Tinjauan-tinjauan teknis adalah salah satu dari banyak tindakan-tindakan yang diperlukan dalam suatu praktik rekayasa perangkat lunak yang baik. Masing-masing tindakan pada dasarnya memerlukan usaha manusia sejumlah tertentu. Karena usaha yang tersedia untuk suatu proyek pada dasarnya bersifat terbatas, merupakan hal yang sangat penting bagi organisasi-organisasi perangkat lunak untuk memahami efektivitas dari masing-masing tindakan dengan mendefinisikan sejumlah metrik yang dapat digunakan untuk melakukan penilaian atas kelayakan masing-masing tindakan itu.

Meski banyak metrik dapat didefinisikan untuk digunakan sebagai sarana bantu untuk tinjauan-tinjauan teknis, sejumlah kecil metrik dapat menyediakan wawasan yang bermanfaat. Perhitungan metrik yang sederhana cukup membantu dalam proses peninjauan. Metrik-metrik tinjauan teknis berikut ini dapat dikumpulkan untuk masing-masing tinjauan teknis yang dilakukan.

- a. Usaha persiapan, E_p
- b. Usaha penilaian, E_a
- c. Usaha pengerjaan ulang, E_r
- d. Ukuran produk kerja, WPS

- e. Kesalahan-kesalahan kecil yang ditemukan, Err_{minor}
- f. Kesalahan-kesalahan besar yang ditemukan, Err_{mayor}

Metrik-metrik di atas selanjutnya bisa diperhalus dengan cara menghubungkannya dengan jenis produk kerja yang telah ditinjau untuk hal apa metrik-metrik itu dikumpulkan.

2.3.2 Menganalisis Metrik-metrik

Sebelum analisis dapat dimulai, suatu perhitungan matematika sederhana harus dilakukan. Usaha peninjauan total dan jumlah total kesalahan-kesalahan yang disingkapkan dalam proses peninjauan seringkali didefinisikan sebagai berikut.

$$E_{review} = E_p + E_a + E_r \quad (2.1)$$

$$Err_{tot} = Err_{minor} + Err_{mayor} \quad (2.2)$$

Sementara itu kerapatan kesalahan (*error density*) yang merepresentasikan kesalahan-kesalahan yang ditemukan untuk tiap unit kerja produk yang ditinjau didefinisikan menggunakan perhitungan metrik *error density* sederhana berikut ini.

$$Error\ density = \frac{Err_{tot}}{WPS} \quad (2.3)$$

Hasil dari E_{review} dan *error density* akan semakin baik ketika nilai yang didapatkan semakin sedikit. Contohnya *error density* 0,1 lebih baik dibandingkan dengan *error density* 0,5 dan E_{review} dengan hasil 2 jam lebih baik dibandingkan dengan 3 jam.

2.3.3 Tinjauan-tinjauan Informal

Tinjauan-tinjauan informal mencakup di dalamnya pemeriksaan sederhana produk kerja rekayasa perangkat lunak oleh rekayasawan-rekayasawan perangkat lunak atas hasil kerja rekan-rekannya, termasuk di dalamnya pertemuan (yang melibatkan lebih dari 2 orang) yang bertujuan untuk melakukan peninjauan sebuah produk kerja perangkat lunak, atau untuk melakukan peninjauan aspek-aspek berorientasi tinjauan dari kode-kode program yang dihasilkan dari tahapan pemrograman perangkat lunak.

Pemeriksaan sederhana atau sebuah pertemuan yang dilakukan oleh rekayasawan perangkat lunak dengan rekan-rekan kerjanya pada dasarnya juga merupakan sebuah tinjauan. Meski demikian, karena tidak adanya perencanaan atau persiapan yang matang, tidak ada

agenda atau struktur pertemuan, dan tidak ada tindak lanjut pada kesalahan-kesalahan yang tidak tersingkap, efektivitas peninjauan perangkat lunak ini akan lebih rendah dibandingkan dengan pendekatan-pendekatan yang lebih formal. Meski demikian, pemeriksaan sederhana dapat menyingkapkan kesalahan-kesalahan yang mungkin akan dijalarkan ke proses-proses perangkat lunak selanjutnya.

2.3.4 Tinjauan-tinjauan Teknis Yang Bersifat Formal

Suatu tinjauan teknis yang bersifat formal (*Formal Technical Review* [FTR]) pada dasarnya merupakan suatu aktivitas kendali kualitas yang dilakukan oleh rekayasawan-rekayasawan perangkat lunak (dan oleh orang-orang lainnya). Sasaran *technical review* sesungguhnya adalah:

- 1) Untuk menyingkapkan kesalahan-kesalahan dalam fungsi-fungsi, logika-logika, atau dalam implementasi setiap representasi suatu perangkat lunak
- 2) Untuk melakukan verifikasi apakah perangkat lunak yang ditinjau sesuai dengan spesifikasi kebutuhan yang telah ditentukan sebelumnya
- 3) Untuk memastikan bahwa perangkat lunak telah direpresentasikan dengan cara yang sesuai dengan standar-standar yang telah ditentukan sebelumnya
- 4) Untuk mencapai pengembangan perangkat lunak yang seragam
- 5) Untuk membuat proyek-proyek perangkat lunak yang lebih mudah untuk dikelola

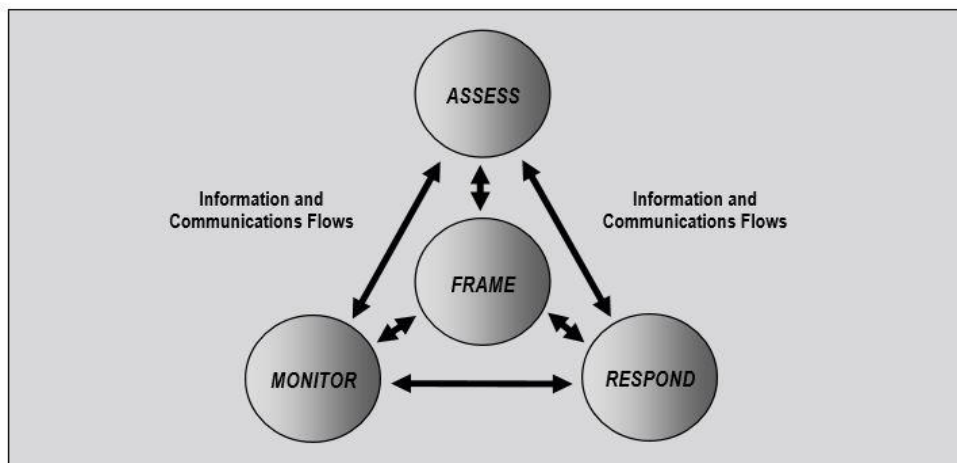
Technical review juga bertindak sedemikian rupa sehingga rekayasawan perangkat lunak bisa menggunakan penyalinan (*backup*) demi kontinuitas pekerjaannya. *Technical review* sesungguhnya merupakan kelas peninjauan teknis yang mencakup di dalamnya penelusuran dan inspeksi. Masing-masing *technical review* dapat dilakukan dengan pertemuan-pertemuan peninjauan dan hal ini baru akan menghasilkan sesuatu yang baik jika masing-masing *technical review* direncanakan, dikendalikan, dan diperhatikan dengan baik pula.

2.4 Konsep Management Risiko

Risiko merupakan kombinasi dari probabilitas suatu kejadian dan konsekuensi dari kejadian tersebut, dengan tidak menutup kemungkinan bahwa ada lebih dari satu konsekuensi untuk satu kejadian, dan konsekuensi bisa merupakan hal yang positif maupun negatif. (Shortreed, 2003). Namun risiko pada umumnya dipandang sebagai sesuatu yang negatif, seperti kehilangan, bahaya, dan konsekuensi lainnya. Kerugian tersebut sebenarnya merupakan bentuk ketidakpastian yang seharusnya dipahami dan dikelola secara efektif oleh organisasi

sebagai bagian dari strategi sehingga dapat menjadi nilai tambah dan mendukung pencapaian tujuan organisasi.

Manajemen risiko didefinisikan sebagai proses, mengidentifikasi, mengukur dan dalam hal ini manajemen risiko akan melibatkan proses-proses, metode dan teknik yang membantu manajer proyek maksimumkan probabilitas dan konsekuensi *event* yang berlawanan. Dalam manajemen proyek, yang dimaksud dengan manajemen risiko proyek adalah seni dan ilmu untuk mengidentifikasi, menganalisis, dan merespon risiko selama umur dan tetap menjamin tercapainya tujuan proyek. Gambar proses manajemen risiko dapat dilihat di Gambar 2.1. Gambar diambil dari sumber (NIST, 2012).



Gambar 2.1 Proses Manajemen Risiko (NIST, 2012)

BAB III METODOLOGI

3.1 Prosedur-prosedur Pengambilan Data

Metode penelitian yang digunakan yaitu metode kualitatif dengan pendekatan studi kasus. Proses pengumpulan data pada penelitian ini dilakukan dengan cara wawancara, observasi, dan kajian pustaka. Pengkajian utama dalam manajemen risiko sistem informasi ini menggunakan kerangka kerja NIST SP 800-30 sebagai panduan prosedur penilaian risiko bidang keamanan informasi. Selain itu juga dilakukan sebuah *technical review* untuk memantau aplikasi sudah terbebas atau belum dari *error* sebelum meluncurkan aplikasi.

Proses manajemen risiko terdapat 3 tahapan yaitu penilaian risiko (*risk assessment*), peringanan risiko (*risk mitigation*), dan evaluasi risiko (*risk evaluation*). Proses penilaian risiko NIST SP 800-30 terdiri dari 4 langkah, meliputi *Prepare for the Assessment*, *Conduct the Assessment*, *Communicate assessment results* dan *Maintain the Assessment*. Prosedur yang dilakukan penulis dalam menentukan Sistem Informasi Penilaian Risiko Bidang Keamanan Informasi dapat dilihat dalam Tabel 3.1.

Tabel 3.1 Prosedur Pengambilan Data

| Teknik | Data | Sumber | Waktu |
|---------------|-------------------------------------------------------------------------|----------------------------|---------------------|
| Wawancara | <i>Programmer</i> diberikan pertanyaan untuk dijawab | Pembuat Aplikasi. | Mei - November 2017 |
| Observasi | <i>Programmer</i> dipantau ketika sedang mengerjakan aplikasi | Pembuat Aplikasi | Mei - November 2017 |
| Studi pustaka | <i>Programmer</i> diminta untuk memahami konsep <i>technical review</i> | Buku, jurnal dan internet. | Mei – November 2017 |

3.2 Technical Review

Di dalam penelitian ini akan diterapkan informal *technical review* dan formal *technical review* (FTR). Informal *technical review* dilakukan dengan memberikan pertanyaan-pertanyaan singkat yang digunakan untuk menemukan kesalahan di dalam aplikasi. Selain itu juga informal *technical review* dapat dilakukan tanpa perencanaan yang khusus dan dialkukan dengan lebih sederhana. Pertanyaan informal *technical review* contohnya adalah “Apakah perancangan

tampilan antarmuka menggunakan konvensi-konvensi yang bersifat baku? Kiri ke kanan? Atas ke bawah?.” Pertanyaan lainnya dapat disesuaikan dengan aplikasi yang akan ditinjau.

Formal *technical review* dilakukan dengan lebih sistematis dibandingkan dengan informal *technical review*. Terdapat 3 point penting di dalam formal *technical review* (Presman, 2010), yaitu pertemuan-pertemuan peninjauan, pelaporan hasil peninjauan dan pemeliharaan catatan-catatan hasil peninjauan, dan panduan-panduan peninjauan.

3.2.1 Pertemuan-pertemuan Peninjauan

Setiap peninjauan perangkat lunak seharusnya mengikuti batasan-batasan berikut ini.

- a. Suatu tinjauan teknis seharusnya melibatkan 3 hingga 5 orang (yang paling umum)
- b. Persiapan yang mendalam boleh saja dilakukan, namun masing-masing orang sebaiknya tidak menghabiskan waktu lebih dari 2 jam kerja untuk melakukannya
- c. Lamanya waktu pertemuan peninjauan teknis sebaiknya kurang dari 2 jam saja

Dengan melihat pada batasan-batasan peninjauan produk kerja perangkat lunak di atas, dapat terlihat bahwa suatu *technical review* sesungguhnya berfokus pada bagian yang spesifik (dan kecil) dari suatu perangkat lunak secara keseluruhan.

3.2.2 Pelaporan Hasil Peninjauan dan Pemeliharaan Catatan-catatan Hasil Peninjauan

Selama proses peninjauan, seorang peninjau produk-produk kerja perangkat lunak yang berperan sebagai perekam akan secara aktif merekam semua permasalahan-permasalahan yang ditemukan. Permasalahan-permasalahan pada produk-produk kerja perangkat lunak itu diikhtisarkan pada bagian akhir pertemuan peninjauan, dan daftar permasalahan yang dihasilkan dari peninjauan dihasilkan. Tambahnya, sebuah laporan kesimpulan peninjauan teknis yang bersifat formal diselesaikan. Suatu laporan kesimpulan peninjauan produk-produk kerja perangkat lunak pada dasarnya menjawab 3 pertanyaan berikut ini.

- a. Apa yang ditinjau?
- b. Siapa yang melakukan peninjauan?
- c. Permasalahan-permasalahan apa yang ditemukan dan bagaimana cara-cara untuk mengatasinya?

Laporan kesimpulan peninjauan produk-produk kerja perangkat lunak seringkali berupa formulir satu halaman (dengan beberapa lampiran). Laporan kesimpulan peninjauan selanjutnya akan menjadi bagian dari rekaman proyek secara historis dan mungkin dapat disebarkan ke pimpinan proyek dan pihak-pihak lain yang terkait. Daftar permasalahan-permasalahan hasil peninjauan produk-produk kerja perangkat lunak memiliki 2 manfaat: (1) untuk

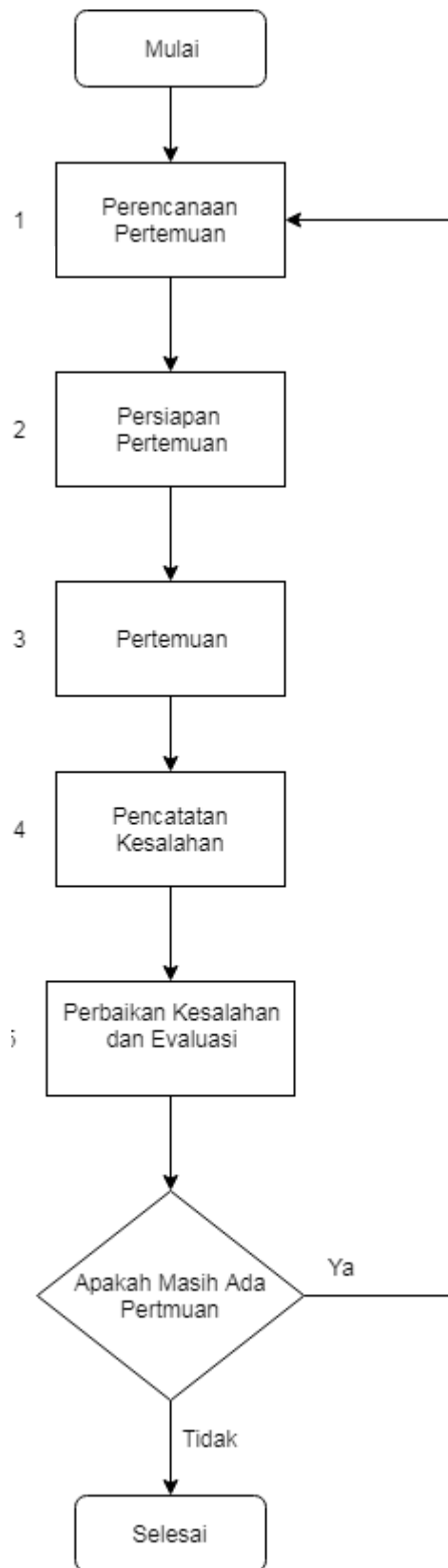
mengidentifikasi area-area permasalahan di dalam perangkat lunak, dan (2) bertindak sebagai daftar periksa atas item-item aksi yang akan memandu produsen saat pembetulan produk.

3.2.3 Panduan-panduan Peninjauan

Panduan-panduan untuk melaksanakan tinjauan-tinjauan teknis secara formal harus ditetapkan dengan cara yang lebih seksama, disebarakan ke semua peninjau, disetujui oleh semua pihak, dan kemudian diikuti oleh siapa pun yang terlibat dalam proses peninjauan produk-produk kerja perangkat lunak secara teknis. Berdasarkan Gambar 3.1 merepresentasikan sejumlah panduan minimum untuk suatu peninjauan teknis yang bersifat formal.

1. Perencanaan Pertemuan
 - a. Batasi jumlah peserta pertemuan dan lakukan persiapan yang memadai.
 - b. Pelatihan yang tepat akan menghasilkan tinjauan yang lebih akurat.
2. Persiapan pertemuan
 - a. Mengatur agenda dan memeliharanya.
 - b. Siapkan daftar periksa untuk masing-masing produk kerja yang akan dinilai
 - c. Alokasikan sumber daya waktu pertemuan
3. Pertemuan
 - a. Peninjauan dilakukan terhadap produk-produk kerja perangkat lunak, tanpa melihat siapa yang membuatnya. Hindari melakukan peninjau bersifat subjektif.
4. Pencatatan kesalahan
 - a. Batasi debat-debat dan ketidaksetujuan. Debat yang berkepanjangan tidak baik dan hanya akan membuang waktu secara sia-sia.
 - b. Perjelas area-area permasalahan, untuk mempermudah proses koreksi masalah yang ditemukan.
 - c. Lakukan pencatatan untuk menghindari kesalahan yang ditemukan tidak ada yang terlewatkan dan semua dapat diselesaikan
5. Perbaiki kesalahan dan evaluasi
 - a. peninjau maupun pembuat aplikasi dapat melihat peninjauan-peninjauan sebelumnya untuk memastikan bahwa kesalahan yang sama tidak terulang kembali.

Setelah ke 5 poin selesai dilakukan. Dilakukan pengecekan apakah masih ada pertemuan yang harus dilakukan atau tidak. Jika masih ada maka kembali dilakukan poin satu hingga poin lima. Jika sudah tidak ada pertemuan maka proses *technical review* telah selesai. Diagram alir *technical review* dapat dilihat pada Gambar 3.1



Gambar 3.1 Proses *Technical Review*

3.3 *Technical Review* Dalam Fase Pengembangan Perangkat Lunak

Di dalam fase pengembangan perangkat lunak terdapat empat poin yang akan di tinjau yaitu Analisis Kebutuhan, desain sistem, desain *database*, dan desain *interface*.

3.3.1 Analisis Kebutuhan

Sebelum pembuatan aplikasi dimulai pastinya harus dilakukan analisis kebutuhan. Analisis kebutuhan memungkinkan pengembang membangun model-model yang akan diterjemahkan ke dalam data, arsitektur, antarmuka, dan prosedural perancangan menjadi perancangan perangkat lunak. Untuk mengetahui apa saja yang dibutuhkan dalam perangkat lunak akan digunakan pertanyaan sebagai berikut.

1. Aplikasi apa yang akan dibuat?
2. Apa manfaat dari aplikasi tersebut?
3. Siapa yang akan menggunakan aplikasi?
4. Apa saja spesifikasi minimal aplikasi yang akan dibuat?
5. Apa kelebihan dari aplikasi ini dibandingkan dengan aplikasi lain yang sejenis?
6. Rencana apa yang telah dibuat untuk penggunaan aplikasi ke depannya?

3.3.2 Desain Sistem

Desain sistem dirancang sebagai penggambaran model sistem untuk mendokumentasikan aspek teknis dan implementasi dari sebuah sistem yang akan dibangun. Pada tahap ini, akan dibuat desain sistem yang akan dibangun, meliputi rancangan sistem, perancangan *database*, dan perancangan *interface* sistem. Perancangan sistem akan dibuat menggunakan *Undefined Model Language* (UML). UML telah menjadi sebuah standar dalam merancang dan mendokumentasikan sebuah sistem. Berikut hasil identifikasi aktor menurut informasi yang telah didapat.

Di dalam desain sistem ini akan dibuat 2 buah UML yaitu:

a. *Use Case Diagram* (UC)

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case Diagram* di aplikasi ini memiliki 3 aktor yaitu Admin, Leader Project, dan Member Project. Setiap aktor memiliki tugasnya masing-masing. Setiap aktor tidak boleh memiliki tugas yang semuanya sama. *Use Case* memiliki, 10 *Use case*, yaitu:

1. *Manajemen Login*
2. *Manajemen User*
3. *Add Project*
4. *Register User*
5. *Edit Project*
6. *Add Member*
7. *Prepare for assessment*
8. *Conduct Assessment*
9. *View Risk Assessment Results*
10. *Setting User*

b. Activity Diagram (AD)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Terdapat beberapa *Activity Diagram* yang akan dibuat, yaitu:

1. *Activity Diagram Manajemen Login*
2. *Activity Diagram Manajemen User*
3. *Activity Diagram Add Project*
4. *Activity Diagram Register User*
5. *Activity Diagram Edit Project*
6. *Activity Diagram Add Member*
7. *Activity Diagram Prepare for Assessment*
8. *Activity Diagram Conduct Assessment*
9. *Activity Diagram View Risk Assessment Results*
10. *Activity Diagram Setting User*

3.3.3 Desain Database (TB)

Desain *Database* dilakukan dengan membuat 2 hal yaitu relasi antar tabel dan struktur tabel. Relasi antar tabel merupakan diagram yang menggambarkan hubungan antara satu tabel dengan tabel lainnya di dalam *database*. Struktur tabel merupakan tabel yang menjelaskan isi dari tabel yang berada di dalam *database*. Tabel yang dibuat akan terdiri dari beberapa kolom

yang di dalamnya terdapat banyak data dengan tipe berbeda-beda. Setiap tabel yang ada berfungsi untuk menyimpan data-data yang nanti akan digunakan perangkat lunak dalam menjalankan fungsinya.

Di dalam projek ini akan terdapat 6 tabel yang terdiri dari:

1. Tabel *User*
2. Tabel *Project*
3. Tabel *User_project*
4. Tabel *ThreatSource*
5. Tabel *ThreatEvent*
6. Tabel *List_event*

3.3.4 Desain *Interface* (UI)

Desain *interface* merupakan sebuah rancangan untuk tampilan dari sebuah perangkat lunak. Perangkat lunak ini akan menggunakan *background* dengan warna dominan putih dan kuning. Desain dapat berubah sewaktu-waktu ketika pembuat aplikasi merasa ada yang kurang dari desain sebelumnya. Terdapat beberapa bagian desain *interface* yang akan dibuat yaitu:

1. Halaman *Login*
2. Halaman *Register*
3. Halaman *Home*
4. Halaman *Confirmation User*
5. Halaman *List User*
6. Halaman *Add Project*
7. Halaman *List Project*
8. Halaman *Detil Project*
9. Halaman *Add Member Project*
10. Halaman *Prepare for Assessment*
11. Halaman *Conduct Assessment (Threat Event, Risk)*
12. Halaman *Result*
13. Halaman *View Details*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Persiapan *Technical Review*

Technical review memiliki beberapa poin-poin penting yang tidak boleh dilewatkan dan harus dilakukan selama kegiatan *technical review* dilakukan. Ketika satu atau lebih poin tidak dilaksanakan dapat mempengaruhi hasil dari *technical review*. Semakin banyak poin yang dilewatkan hasil tinjauan akan semakin berkurang. *Technical review* yang baik akan memberikan dampak positif terhadap biaya dan waktu pembuatan perangkat lunak.

4.1.1 Pertemuan-pertemuan Peninjauan

Sebelum pertemuan dimulai ada beberapa hal yang harus diperiksa sesuai dengan sub bab 3.2, yaitu:

1. Dalam penelitian ini poin pertama tidak dilakukan dengan 3 orang karena tinjauan kali ini hanya dilakukan oleh 2 orang (satu orang peninjau dan satu orang pengembang aplikasi). Namun hal ini tidak menjadi masalah karena proses *technical review* dapat berjalan dengan baik.
2. Persiapan yang dilakukan sebelum pertemuan berhasil terlaksana dengan baik. Sebelum pertemuan dimulai materi yang akan ditinjau sudah dipersiapkan sebelumnya sehingga para peninjau dapat melakukan persiapan secara maksimal.
3. Pertemuan dilakukan dari pukul 13.00 WIB hingga pukul 15.00 WIB. Hal ini dapat dilakukan karena setiap pertemuan hanya membahas poin-poin penting sehingga mengurangi waktu dilakukannya pertemuan. Hal ini dapat dilakukan karena sebelum pertemuan dimulai setiap peninjau sudah mempersiapkan pendalaman materi dengan baik.

4.1.2 Pelaporan Hasil Peninjauan

Untuk pelaporan dilakukan dengan cukup baik karena hasil tinjauan dapat dilihat di sub bab selanjutnya. Namun yang kurang adalah tidak adanya tanggal ketika tinjauan dilakukan karena biasanya tinjauan dilakukan setiap hari saat siang hari kecuali hari libur dan ketika pembuat aplikasi atau peninjau sedang berhalangan untuk melakukan pertemuan. Pelaporan yang ada juga disimpan dengan baik untuk digunakan saat pertemuan masa-masa selanjutnya.

Pelaporan dilakukan oleh satu orang peninjau, pembuat aplikasi tidak membuat laporan peninjauan. contoh formulir pelaporan dapat dilihat pada bagian Lampiran.

4.1.3 Panduan-Panduan Peninjauan

Dalam melakukan *technical review* terdapat beberapa panduan-panduan yang harus dilakukan untuk mendapatkan hasil tinjauan yang maksimal dan objektif. Berdasarkan panduan dari sub bab 3.2 didapatkan hasil sebagai berikut:

1. Jumlah peserta pertemuan hanya berjumlah 2 orang. 1 peninjau dan 1 pembuat perangkat lunak. Sebelum pertemuan dilakukan persiapan sudah maksimal yaitu sebelum pertemuan dilakukan sudah siap poin-poin permasalahan yang akan dibahas sehingga pertemuan yang dilakukan dapat menghasilkan solusi yang tepat.
2. Masing-masing peserta belum memperoleh pelatihan dalam membuat perangkat lunak dan melakukan peninjauan. Hanya bekerja berdasarkan dari referensi yang benar.
3. Agenda pertemuan sudah diatur seminggu sekali saat siang hari di Perpustakaan Pusat UII, namun belum terpelihara dengan baik karena ada 1 minggu tidak dilakukan pertemuan sama sekali karena kesibukan peninjau dan pembuat perangkat lunak.
4. Daftar periksa yang digunakan mengambil dari *review checklist* yang sudah ada dengan beberapa penambahan. Hal ini mengakibatkan daftar periksa lebih cocok digunakan untuk proses pemeriksaan kesalahan setelah proses penambahan *review checklist*
5. Alokasi sumber daya waktu tidak lebih dari 2 jam setiap dilaksanakan sebuah pertemuan. Hal ini sesuai kesepakatan yang sudah dibuat
6. Peninjauan dilakukan terhadap sebuah produk yaitu Sistem Penilaian Risiko Dengan Standar NIST 800-30 (Ikhsani, 2017). Pertemuan dilakukan tidak memandang siapa yang membuatnya, hanya fokus terhadap hasil kerjanya saja.
7. Dalam proses pencarian kesalahan terdapat debat namun tidak terlalu lama sehingga proses pencatatan dapat dilanjutkan. Setiap debat yang terjadi dapat diselesaikan dengan baik. Debat yang terjadi tidak mempengaruhi waktu pelaksanaan pertemuan peninjauan.
8. Setiap ditemukan masalah dilakukan penjelasan kenapa masalah tersebut penting untuk diselesaikan dan cara penyelesaiannya. Hal ini memperjelas apa saja yang harus dilakukan pembuat perangkat lunak dalam membenarkan kesalahan yang ditemukan. Perbaiki kesalahan menjadi lebih tepat dan efisien.
9. Catatan setiap kesalahan yang ditemukan sebagai referensi untuk proses peninjauan selanjutnya. Jangan sampai ada kesalahan yang luput dari proses pencatatan.
10. Setiap melakukan peninjauan, peninjau selalu melihat hasil peninjauan sebelumnya. Hal ini dilakukan untuk mengurangi kemungkinan munculnya kesalahan yang sudah pernah

muncul sebelumnya. Hasil peninjauan yang ada juga dapat digunakan sebagai pedoman untuk melakukan tinjauan masa-masa selanjutnya.

4.2 Laporan Pertemuan *Technical Review*

Pertemuan *technical review* dilakukan di Perpustakaan Terpadu Universitas Islam Indonesia pada jam 13.00 WIB. Dalam 15 pertemuan, rata-rata waktu pelaksanaan tinjauan adalah 3,4 jam dengan rata-rata *error density* 0,17813. Hasil pertemuan berdasarkan waktu dilakukannya pertemuan dapat dilihat pada Tabel 4.1

Tabel 4.1 Hasil Pertemuan *Technical Review* Berdasarkan Waktu

| No | Topik | E _{review} | Error Density |
|----|------------------|---------------------|---------------|
| 1 | Use Case | 2 Jam | 0,4 |
| 2 | Desain Interface | 2 Jam | 0,4 |
| 3 | Use Case | 2 Jam | 0,35 |
| 4 | Activiy Diagram | 2 Jam | 0,07 |
| 5 | Desain Database | 2 Jam | 0 |
| 6 | Desain Interface | 2 Jam | 0,3 |
| 7 | Use Case | 2 Jam | 0,3 |
| 8 | Activity Diagram | 2 Jam | 0,06 |
| 9 | Desain Database | 2 Jam | 0 |
| 10 | Desain Interface | 2 Jam | 0,3 |
| 11 | Use Case | 2 Jam | 0,3 |
| 12 | Activity Diagram | 2 Jam | 0,042 |
| 13 | Desain Database | 2 Jam | 0 |
| 14 | Desain Interface | 2 Jam | 0,15 |
| 15 | Source Code | 3 Jam | 0 |

Jumlah keseluruhan pertemuan yang dilakukan adalah 15 pertemuan dengan 4 pertemuan membahas *use case*, 3 pertemuan membahas *activity diagram*, 3 pertemuan membahas desain *database*, 4 pertemuan membahas desain *interface*, 1 pertemuan membahas *source code*. Setiap Pertemuan yang dilakukan menghasilkan laporan pertemuan. Hasil keseluruhan pertemuan berdasarkan topik dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Pertemuan *Technical Review* Berdasarkan Topik

| No | Topik | Kesalahan Yang Ditemukan | Keterangan |
|----|------------------|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| 1 | Use Case | <i>Association</i> belum ada penunjuk yang tepat, aktor Admin lebih sedikit tugasnya dibandingkan dengan aktor User | <i>Use Case</i> memiliki 2 aktor dan 7 Use Case |
| 2 | | <i>Association</i> belum tepat, bahasa <i>use case</i> masih campur belum serasi | <i>Use case</i> memiliki 4 aktor dan 9 use case |
| 3 | | <i>Association</i> belum tepat, bahasa <i>use case</i> masih kurang tepat | <i>Use case</i> memiliki 4 aktor dan 10 use case |
| 4 | | Bahasa masih kurang tepat | <i>Use case</i> emiliki 3 aktor dan 11 use case |
| 5 | Activity Diagram | Bahasa masih kurang tepat, <i>decision</i> masih kurang tepat | <i>Activity</i> diagram berjumlah 8 |
| 6 | | Bahasa masih kurang tepat, <i>decision</i> masih kurang tepat | <i>Activity</i> diagram berjumlah 9 |
| 7 | | Bahasa masih kurang tepat, <i>decision</i> masih kurang tepat | <i>Activity</i> diagram berjumlah 10 |
| 8 | Desain | Tidak ditemukan kesalahan | Jumlah tabel sebanyak 11 |
| 9 | Database | Tidak ditemukan kesalahan | Jumlah tabel sebanyak 12 |
| 10 | | Tidak ditemukan kesalahan | Jumlah tabel sebanyak 6 |
| 11 | Desain Interface | Keseluruhan fungsi sistem belum digambarkan. Warna tampilan terlalu <i>monochrom</i> | Terdapat 26 desain <i>interface</i> , warna utama hitam abu-abu |
| 12 | | Kebutuhan sistem belum tergambarkan semua, warna masih <i>monochrom</i> | Terdapat 22 desain <i>interface</i> , warna utama hitam abu-abu |
| 13 | | Fungsi sistem masih belum tergambarkan semua | Terdapat 11 desain <i>interface</i> , warna utama putih dan orange |
| 14 | | Masih ada fungsi sistem yang belum digambarkan | Terdapat 15 desain <i>interface</i> , warna utama putih dan orange |
| 15 | Source Code | Tidak ditemukan kesalahan | Ukuran folder Source code berjumlah 29,4 MB. Memiliki 15 kelas dan 260 variabel. |

4.3 Hasil *Technical Review*

Technical review dilakukan dalam periode 6 bulan. Proses pembuatan aplikasi ditinjau dari awal pembuatan hingga aplikasi selesai dibuat. Terdapat beberapa hal yang dilakukan *review* yaitu:

1. Analisis Kebutuhan
2. *Use Case* Diagram
3. *Activity* Diagram
4. Desain *Database*
5. Desain *Interface*
6. *Source Code*

Setiap poin dilakukan tinjauan dengan rentang waktu dan tanggal yang berbeda. Untuk melakukan *technical review* satu pertemuan dapat membahas lebih dari satu permasalahan,

selama waktu yang ditetapkan masih ada. Tergantung dari kompleksitas masalah, pertemuan dapat dilakukan lebih dari 1 kali untuk membahas 1 buah point.

4.3.1 Analisis Kebutuhan

Analisis kebutuhan dari sub bab 3.3 digunakan sebagai patokan dalam membuat keseluruhan dari aplikasi. Aplikasi yang akan dibuat adalah “Sistem Penilaian Risiko Bidang Keamanan Informasi pada Sistem Informasi Menggunakan National *Institute of Standard and Technology* (NIST) SP 800-30.” Perangkat lunak ini bermanfaat untuk membantu memudahkan tim auditor dalam melakukan proses penilaian risiko pada sistem informasi bidang keamanan informasi. Tim auditor akan lebih dalam melakukan rekap penilaian risiko pada sistem informasi bidang keamanan informasi.

Pengguna dari sistem ini adalah Auditor yang bertugas untuk menilai risiko bidang keamanan informasi pada sistem informasi perusahaan. Tim auditor terdiri dari leader project dan member project. Spesifikasi minimal dari aplikasi dibagi menjadi 2 yaitu *hardware* dan *software*:

a. *Hardware*

Spesifikasi *hardware* yang digunakan adalah sebagai berikut:

Processor : Intel Core i3-3217U 1.80 GHz

Memory : 6 GB

Harddisk : 500 GB

b. *Software*

Spesifikasi *software* yang digunakan adalah sebagai berikut:

Sistem Operasi : Windows 10 Pro

Bahasa Program : PHP, HTML, JavaScript, CSS

Tools : Sublime Text 3, Code Igniter 3, Bootstrap 4

DBMS : My SQL Server

Web Server : Apache

Web Browser : Google Chrome

Kelebihan dari aplikasi ini dibandingkan dengan aplikasi lain adalah lebih mudah digunakan dibandingkan dengan menggunakan langsung standar NIST. Setiap pengguna sistem memiliki peran yang jelas dan saling berhubungan sehingga setiap penilaian risiko yang

dilakukan dapat dilaksanakan secara lebih efisien. Untuk rencana ke depannya saat ini belum ada karena jika ingin dikembangkan ke aplikasi mobile dikhawatirkan akan menambahkan risiko kebocoran informasi perusahaan. Karena itulah untuk saat ini masih fokus dikembangkan hanya dalam bentuk *website*.

4.3.2 Use Case Diagram

Use case diagram di dalam aplikasi memiliki 3 aktor yaitu Admin, Leader Project, dan Member Project. Admin dapat melakukan Manajemen Login, Manajemen *User*, dan *Add Project*. Leader Project dapat melakukan *Register User*, *Edit Project*, *Add Member*, *Prepare for Assessment*, *Conduct Assessment*, dan *View Risk Assissment Results*, dan *Setting User*. Member Project dapat melakukan *Register User*, *Prepare for Assessment*, *Conduct Assessment*, *View Risk Assessment Results*, dan *Setting User*. *Review checklist* diambil dari sumber (IT Toolbox, 2007) dengan beberapa modifikasi. Template untuk *review checklist use case* dapat dilihat pada Lampiran A. Hasil *review Use Case Diagram* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil *Review Use Case Diagram*

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|-------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Apakah setiap <i>use case</i> secara tepat menyatakan satu atau lebih kebutuhan fungsional yang sistem harus penuhi? | Ya | Setiap <i>use case</i> sudah menyatakan semua kebutuhan fungsional sistem |
| 2 | Apakah Setiap <i>use case</i> yang ada sudah sesuai standar? | Ya | Setiap <i>use case</i> yang ada sudah berbentuk oval bukan lingkaran dan berupa kata kerja |
| 3 | Apakah setiap aktor sudah tepat sesuai aturan penamaan aktor? | Ya | Setiap Aktor yang dibuat sudah sesuai dengan aturan. Tidak ada duplikasi aktor. |
| 4 | Apakah kumpulan <i>use case</i> menangkap semua kebutuhan fungsional dari sistem? | Tidak | Salah satu <i>use case</i> masih kurang tepat di bagian <i>assosiacion</i> |
| 5 | Apakah <i>use case</i> spesifik sehingga tidak ambigu, dan hanya memiliki satu tafsiran? | Tidak | <i>Use case</i> masih menggunakan bahasa campuran Indonesia dan Inggris. |
| 6 | Apakah kumpulan <i>use case</i> konsisten secara internal, sehingga tidak ada subset <i>use case</i> konflik dengan subset lainnya? | Tidak | <i>Use case</i> belum konsisten dalam segi penggunaan bahasa masih terdapat campuran 2 bahasa yang berbeda |
| 7 | Apakah kumpulan <i>use case</i> juga konsisten secara eksternal dengan produk OOAD lainnya? | Ya | Untuk bagian <i>activity diagram</i> sudah konsisten dengan <i>use case</i> , tidak terdapat perbedaan dari jumlah <i>use case</i> dengan <i>activity diagram</i> |
| 8 | Spesifikasi produk dapat dilakukan pengujian untuk menentukan terpenuhi atau tidaknya kebutuhan dari <i>use case</i> | Ya | Apakah terdapat proses hemat biaya untuk menentukan bahwa produk aplikasi memenuhi |

| | | | |
|----|----------------------------------------------------------------------------------------------------------------------|----|----------------------------------------------------------------------------------------------------------------------------------------------|
| | | | kebutuhan yang dinyatakan oleh use case? |
| 9 | Apakah struktur dan gaya <i>use case</i> mendukung perubahan yang dapat dibuat dengan mudah, lengkap, dan konsisten? | Ya | Struktur <i>use case</i> sudah dibuat untuk memungkinkan dilakukannya perubahan ketika terjadi sebuah kesalahan |
| 10 | Apakah <i>use case</i> mendukung penelusuran ke belakang dan depan? | Ya | <i>Use case</i> sudah mendukung penelusuran ke belakang yaitu dari bagian analisis kebutuhan dan juga ke depan yaitu <i>activity diagram</i> |

Metriks untuk Use Case Diagram

Usaha persiapan, $E_p = 1$ jam

Usaha penilaian, $E_a = 1$ jam

Usaha pengerjaan ulang, $E_r = 1$ jam

$E_{\text{review}} = E_p + E_a + E_r = 1 + 1 + 1 = 3$ Jam

Kesalahan total, $Err_{\text{tot}} = 3$

Ukuran produk kerja, $WPS = 10$ (berdasarkan total dari *checklist* tinjauan *use case*)

$Error\ Density = \frac{Err_{\text{tot}}}{WPS} = \frac{3}{10} = 0,3$

Berdasarkan rumus 2.1 dapat ditarik kesimpulan bahwa total jam yang dibutuhkan untuk melakukan *review* adalah 3 jam, artinya tinjauan telah dilakukan dengan baik. Dengan menggunakan rumus 2.2, kesalahan total ditemukan sebanyak 3 kesalahan. WPS berjumlah 10 berdasarkan jumlah *review checklist*. Dengan menggunakan rumus 2.3 ditemukan kerapatan kesalahan (*error density*) yang terjadi adalah 0,3. Hal ini menandakan kira-kira kemungkinan terjadi kesalahan adalah sepertiga dari sebuah bagian produk.

4.3.3 Activity Diagram

Activity diagram menggambarkan aktivitas yang dilakukan di dalam sistem. Aktor dan sistem digambarkan di dalam *swimlanes*. Di dalam aplikasi ini terdapat 9 *activity diagram*, yaitu:

1. Activity Diagram Manajemen Login
2. Activity Diagram Manajemen *User*
3. Activity Diagram *Add Project*
4. Activity Diagram *Register User*
5. Activity Diagram *Edit Project*
6. Activity Diagram *Add Member*
7. Activity Diagram *Prepare for Assessment*

8. Activity Diagram *Conduct Assessment*
9. Activity Diagram *View Risk Assessment*
10. Activity Diagram *Setting User*

Sumber tabel untuk *review checklist* berasal dari (Allaboutrequirements, 2011) dengan beberapa modifikasi. Template *review checklist activity diagram* dapat dilihat pada Lampiran B. Hasil *review* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil *Review Activity Diagram*

| No | Item | Activity Diagram | | | | | | | | | |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------|---|---|---|---|---|---|---|---|----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | Apakah activity diagram memiliki paling tidak satu <i>node</i> awal? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 2 | Apakah activity diagram memiliki paling tidak satu <i>node</i> akhir? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 3 | Apakah activity diagram memiliki aktor <i>swimlanes</i> dan sistem <i>swimlanes</i> ? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 4 | Apakah setiap aktivitas direpresentasikan dengan bentuk kotak? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 5 | Apakah nama <i>Swimlanes</i> sudah benar? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 6 | Apakah situasi <i>error</i> teratasi? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 7 | Apakah sudah dipastikan bahwa <i>activity diagram</i> dapat muat di 1 atau 2 halaman (jika tidak dapat mengindikasikan detail label yang salah)? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 8 | Apakah sudah pasti setiap aktivitas aktor <i>swimlanes</i> , adalah cakupan dari layar GUI? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 9 | Apakah sudah dipastikan alur aktivitas tidak diinterupsi? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 10 | Apakah terdapat desain navigasi dari <i>use case</i> ? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 11 | Apakah jelas kapan kegunaan <i>use case</i> digunakan? | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| 12 | Apakah Penggunaan <i>decision</i> sudah tepat? | Y | T | Y | Y | T | T | T | T | Y | Y |

Ket : Y = Ya

T = Tidak

Kesalahan terjadi pada *activity diagram* nomor 2, 5, 6, 7, dan 8. *Activity diagram* yang salah terjadi pada bagian *decision*. Penempatan *True* and *False* masih kurang tepat dari segi posisi dan bahasa menggunakan bahasa Inggris sedangkan *action* yang lain menggunakan bahasa Indonesia. Selain kesalahan di atas tidak ada kesalahan fatal yang muncul, hanya kesalahan-kesalahan kecil saja yang ada di bagian *activity diagram*

Metriks untuk *Activity Diagram*

Usaha persiapan, $E_p = 1$ jam

Usaha penilaian, $E_a = 1$ jam

Usaha pengerjaan ulang, $E_r = 1$ jam

$E_{review} = E_p + E_a + E_r = 1 + 1 + 1 = 3$ Jam

Kesalahan total, $Err_{tot} = 5$

Ukuran produk kerja, $WPS = 120$ (berdasarkan total dari *Review checklist*)

$$Error\ Density = \frac{Err_{tot}}{WPS} = \frac{5}{120} = 0,042$$

Berdasarkan rumus 2.1, jumlah jam dari usaha persiapan, usaha penilaian, dan usaha pengerjaan ulang adalah sebesar 3 jam yang merupakan total dari usaha *review*. Lamanya waktu untuk melakukan *review* yaitu 3 jam terhitung cepat dalam artian telah dilakukan dengan baik. Dengan menggunakan rumus 2.2 ditemukan kesalahan total sebanyak 5 kesalahan. WPS berjumlah 120 dihitung dari jumlah *review checklist* yang ada. Dengan menggunakan rumus 2.3 ditemukan kerapatan kesalahan (*error density*) dari *activity diagram* sebesar 0,042 kesalahan untuk setiap *activity diagram*. Ini menandakan terdapat sedikit sekali kemungkinan terjadinya kesalahan pada bagian *activity diagram*.

4.3.4 Desain Database

Database di dalam aplikasi ini memiliki 6 tabel, yaitu tabel *user*, tabel *project*, tabel *user_project*, tabel *threatSource*, tabel *threatEvent*, dan tabel *list_event*. Setiap tabel memiliki datanya sendiri-sendiri dan saling berhubungan satu sama lain. Sumber dari *review checklist* untuk desain *database* berasal dari (Murali, 2010) dengan beberapa modifikasi. Template *review checklist* desain *database* dapat dilihat pada Lampiran C. Tabel hasil tinjauan dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil *Review* Desain *Database*

| No | Item | Ya/Tidak | Keterangan |
|----|----------------------------------------------------------------------------------------------|----------|----------------------------------------------------------------------------|
| 1 | Apakah setiap entitas diubah menjadi tabel? | Ya | Setiap entitas yang ada telah diubah menjadi tabel |
| 2 | Apakah setiap tabel sudah terdapat indeks? | Ya | Setiap tabel sudah memiliki <i>primary key</i> sebagai index tabel |
| 3 | Sudahkah setiap <i>foreign key</i> di spesifikasikan? | Ya | Setiap <i>foreign key</i> sudah spesifik untuk setiap tabel |
| 4 | Apakah integritas data sesuai dengan aturan integritas data database management sistem? | Ya | Data sudah sesuai dengan aturan integritas data database management sistem |
| 5 | Apakah privileges dan autoritas, data akses kontrol, dan lain-lain. Sudah diimplementasikan? | Ya | Privileges dan autoritas sudah diimplementasikan di dalam database |

| | | | |
|-----------|----------------------------------------------------------------|-----------------|---------------------------------------------------------------------------|
| 6 | Apakah nama tabel sudah benar? | Ya | Nama tabel dalam database sudah sesuai dengan aturan penulisan nama tabel |
| 7 | Apakah nama kolom data sudah benar? | Ya | Nama kolom di dalam tabel sudah benar dan sesuai aturan penamaan kolom |
| 8 | Apakah tipe data sudah benar? | Ya | Setiap tipe data sudah sesuai dengan isi dari data yang nanti akan diisi |
| 9 | Apakah penggunaan <i>foreign key</i> sudah tepat? | Ya | Penggunaan <i>foreign key</i> untuk setiap tabel sudah tepat |
| No | Item | Ya/Tidak | Keterangan |
| 10 | Apakah setiap tabel memiliki <i>primary key</i> ? | Ya | Setiap tabel sudah memiliki <i>primary key</i> |
| 11 | Apakah penggunaan <i>constraint</i> sudah tepat? | Ya | Penggunaan <i>constraint</i> untuk setiap tabel sudah tepat |
| 12 | Apakah bahasa SQL tidak digunakan sebagai nama objek database? | Ya | Tidak ada bahasa SQL yang digunakan untuk menamai objek di dalam database |

Metriks untuk Desain Database

Usaha persiapan, $E_p = 1$ jam

Usaha penilaian, $E_a = 1$ jam

Usaha pengerjaan ulang, $E_r = 1$ jam

$E_{review} = E_p + E_a + E_r = 1 + 1 + 1 = 3$ Jam

Kesalahan total, $Err_{tot} = 0$

Ukuran produk kerja, $WPS = 12$ (berdasarkan total dari *Review checklist*)

$$Error\ Density = \frac{Err_{tot}}{WPS} = \frac{0}{12} = 0$$

Berdasarkan rumus 2.1 Didapatkan waktu pelaksanaan *review* adalah selama 3 jam. Jumlah jam termasuk ke dalam kategori baik dan tepat untuk melakukan *review*. Rumus 2.2 dan rumus 2.3 menghasilkan jumlah yang sama yaitu 0, artinya tidak terdapat Kesalahan di dalam peninjauan desain *database*.

4.3.5 Desain Interface

Bagian ini akan melakukan *review* terhadap *interface*. *Interface* merupakan bagian dari aplikasi yang pertama kali dilihat oleh user. *Review checklist* diambil dari (fdot, 2013) dengan beberapa modifikasi. *Review interface* memiliki 2 inti pembahasan yaitu *detailed design* dan *user interface design*. *Detailed design* merepresentasikan solusi visual untuk aplikasi dilihat dari pandangan pengguna (No. 1 sampai No. 4). *User interface design* mendefinisikan *user*

interface untuk sistem (No. 5 sampai No. 13). *Template review checklist* desain *Interface* dapat dilihat pada Lampiran D. Hasil *review* Desain *interface* dapat dilihat pada tabel 4.6.

Tabel 4.6 Hasil *Review* Desain *Interface*

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|---------------------------------------------------------------------------------------------------------------------|------------------|------------------------------------------------------------------------------------|
| 1 | Apakah desain dan teknologi untuk pertukaran informasi setiap komponen, fungsi, dan tugas sudah ditetapkan? | Ya | Setiap teknologi di dalam sistem sudah ditetapkan di dalam desain |
| 2 | Apakah desain dan teknologi dapat bertukar informasi antara setiap <i>interface</i> ? | Ya | Setiap <i>interface</i> telah dapat melakukan pertukaran informasi |
| 3 | Apakah semua fitur dan kemampuan yang dibutuhkan sudah dibuat? | Tidak | Semua fitur yang dibutuhkan untuk sistem belum selesai dibuat |
| 4 | Apakah komponen sistem termasuk perangkat keras, jaringan, sistem operasi, dan database yang dibutuhkan ditetapkan? | Ya | Setiap komponen sistem sudah ditetapkan di dalam desain <i>interface</i> |
| 5 | apakah <i>user interface</i> untuk sistem sudah ditetapkan? | Ya | User interface untuk sistem sudah ditetapkan |
| 6 | Apakah hierarki navigasi layar sudah ditetapkan? | Ya | Hierarki navigasi layar sudah ditetapkan |
| 7 | Apakah terdapat mockup layar? | Ya | Mockup layar sudah ada |
| 8 | Apakah informasi atribut yang cocok untuk setiap tampilan sudah di tunjukkan? Apakah keamanan sudah di definisikan? | Ya | Informasi atribut dan keamanan sudah ditetapkan di dalam desain <i>interface</i> |
| 9 | Apakah komponen utama lain dari fungsi pengguna yang membutuhkan interface sudah dapat digunakan? | Ya | Komponen utama lain dari sistem sudah digambarkan di dalam desain <i>interface</i> |
| 10 | Apakah Desain sudah memiliki bagian <i>error handling</i> ? | Tidak | Desain <i>interface</i> belum menggambarkan <i>error handling</i> |
| 11 | Apakah ukuran <i>font</i> tulisan sudah cukup jelas? | Ya | Ukuran <i>font</i> di dalam desain sudah cukup jelas |
| 12 | Apakah posisi setiap menu sudah tepat? | Ya | Posisi setiap menu sudah tepat di dalam <i>desain interface</i> |
| 13 | Apakah tata letak tulisan sudah tepat? | Ya | Tata letak penulisan di dalam <i>desain interface</i> sudah tepat |

Metriks untuk Desain *Interface*

Usaha persiapan, $E_p = 1$ jam

Usaha penilaian, $E_a = 1$ jam

Usaha pengerjaan ulang, $E_r = 1$ jam

$E_{review} = E_p + E_a + E_r = 1 + 1 + 1 = 3$ Jam

Kesalahan total, $Err_{tot} = 2$

Ukuran produk kerja, $WPS = 13$ (berdasarkan total dari desain *interface checklist*)

$Error\ Density = \frac{Err_{tot}}{WPS} = \frac{2}{13} = 0,15$

Berdasarkan rumus 2.1, usaha total dari persiapan, penialaian, dan pengerjaan ulang adalah selama 3 jam. Total jam *review* termasuk ke dalam kategori baik dalam proses formal *technical review*. Berdasarkan rumus 2.2 ditemukan 2 total kesalahan di dalam desain *interface*. Berdasarkan rumus 2.3 ditemukan kerapatan kesalahan (*error density*) untuk bagian desain *interface* adalah 0,15. Hal ini menandakan kemungkinan terjadinya kesalahan kurang dari seperempat.

4.3.6 *Source Code*

Source code merupakan kumpulan dari beberapa bahasa pemrograman untuk menyelesaikan masalah. *Review Checklist* diambil dari (Gutha, 2015) dengan beberapa modifikasi. Template *review checklist source code* dapat dilihat pada Lampiran E. Hasil *review source code* dapat dilihat pada tabel 4.7.

Tabel 4.7 Hasil *Review Source Code*

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|-------------------------------------------------------------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Apakah setiap <i>method</i> menggunakan bahasa yang jelas? | Ya | Bagian <i>controllers</i> memiliki 56 <i>method</i> yang terdapat di dalam 4 file. Bagian <i>models</i> memiliki 53 <i>method</i> yang terdapat di dalam 11 file. Bagian <i>views</i> tidak mengandung <i>method</i> |
| 2 | Apakah setiap baris kode yang ditulis bebas dari typo? | Ya | Baris kode untuk <i>controllers</i> berjumlah 727 baris, 5 <i>file</i> , dengan ukuran folder 24,7 KB. Baris kode untuk <i>models</i> memiliki 366 baris, 13 file, dengan ukuran folder 9,20 KB. Baris <i>views</i> memiliki 2944 baris, 3 sub folder, 2 file, dan ukuran folder 120 KB. |
| 3 | Apakah setiap kelas, variabel, dan <i>method modifiers</i> sudah tepat? | Ya | <i>Source code</i> memiliki 15 kelas yang terdapat pada bagian <i>models</i> dan <i>controllers</i> . <i>Source code</i> memiliki variabel sebanyak 260 variabel. <i>Method modifier</i> yang digunakan di dalam <i>source code</i> adalah <i>public</i> . |

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|----------------------------------------------------------------------------------|------------------|----------------------------------------------------------------------------|
| 4 | Apakah setiap kebutuhan sistem sudah dibuat ke dalam bentuk <i>source code</i> ? | Ya | Setiap fungsi dari sistem sudah selesai dibuat ke dalam <i>source code</i> |
| 5 | Apakah <i>Source code</i> sudah berukuran sesuai dengan kebutuhan sistem? | Ya | Ukuran folder dari <i>source code</i> berjumlah 29,4 MB |

Metriks untuk *Source Code*

Usaha persiapan, $E_p = 1$ jam

Usaha penilaian, $E_a = 2$ jam

Usaha pengerjaan ulang, $E_r = 2$ jam

$E_{review} = E_p + E_a + E_r = 1 + 2 + 2 = 5$ Jam

Kesalahan total, $Err_{tot} = 0$

Ukuran produk kerja, $WPS = 5$ (berdasarkan total dari *Source Code checklist*)

$$Error\ Density = \frac{Err_{tot}}{WPS} = \frac{0}{5} = 0$$

Berdasarkan dari rumus 2.1 dapat disimpulkan bahwa waktu untuk melakukan peninjauan adalah selama 5 jam. Hal ini menandakan tinjauan yang dilakukan berjalan lebih lama dari seharusnya (Kurang dari 3 jam). Rumus 2.2 dan 2.3 menghasilkan perhitungan sebanyak 0, artinya tidak ditemukan kesalahan yang ditemukan dalam tinjauan menggunakan *review checklist*.

4.3.7 Review Pengujian

Review pengujian di dalam laporan ini menggunakan metode *traceability* matriks berdasarkan *use case*. *Traceability* matriks merupakan dokumen (biasanya terdiri dari tabel) yang digunakan untuk melakukan pengecekan sudah terpenuhi atau belum kebutuhan sebuah sistem. Selain itu dilakukan juga pengecekan semua fungsi yang telah dibuat di dalam sistem, apakah sudah berjalan sesuai dengan rencana atau tidak. Menurut (Guru99, 2011) terdapat dokumen penting di dalam *traceability* matriks yaitu *Test Case* (TC). Di dalam TC terdapat 2 kolom yaitu *use case* dan keterangan (Wibowo, 2017). Tabel TC dapat dilihat pada Tabel 4.8.

Tabel 4.8 *Test Case* Aplikasi

| No | Use Case | TC# = Keterangan |
|----|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <i>Login</i> | TC01 = <i>Username</i> tidak boleh kosong TC02 = <i>Password</i> tidak boleh kosong TC03 = Jika <i>username</i> dan <i>password</i> benar, berhasil login |
| 2 | Manajemen <i>User</i> | TC04 = Admin menerima permintaan <i>user</i> TC05 = Admin menolak permintaan <i>user</i> TC06 = Admin menghapus <i>user</i> |
| 3 | <i>Add Project</i> | TC07 = Admin memasukkan nama <i>project</i> TC08 = Admin memasukkan deskripsi <i>project</i> TC09 = Admin memasukkan tanggal mulai <i>project</i> TC10 = Admin memasukkan tanggal <i>deadline project</i> TC11 = Admin memasukkan Leader <i>Project</i> TC12 = Jika data <i>project</i> sudah lengkap, berhasil membuat <i>project</i> baru |
| 4 | <i>Register User</i> | TC13 = Leader <i>Project</i> /Member <i>Project</i> memasukkan <i>username</i> TC14 = Leader <i>Project</i> /Member <i>Project</i> memasukkan <i>email</i> TC15 = Leader <i>Project</i> /Member <i>Project</i> memasukkan <i>password</i> TC16 = Leader <i>Project</i> memilih pilihan Leader <i>Project</i> TC17 = Member <i>Project</i> memilih pilihan Member <i>Project</i> TC18 = Jika data Leader <i>Project</i> /Member <i>Project</i> sudah lengkap, berhasil melakukan pendaftaran |
| 5 | <i>Edit Project</i> | TC19 = Leader <i>Project</i> mengubah nama <i>project</i> TC20 = Leader <i>Project</i> mengubah deskripsi <i>project</i> TC21 = Jika data <i>project</i> sudah lengkap, berhasil melakukan perubahan |
| 6 | <i>Add Member</i> | TC22 = Leader <i>Project</i> memasukkan Member <i>Project</i> baru TC23 = Jika sudah memilih Member <i>Project</i> baru, Member <i>Project</i> masuk ke dalam <i>project</i> TC24 = Leader <i>Project</i> menghapus Member <i>Project</i> |
| No | Use Case | TC# = Keterangan |

| | | |
|----|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 | <i>Prepare for Assessment</i> | <p>TC25 = Leader Project/Member Project memasukkan tujuan penilaian</p> <p>TC26 = Leader Project/Member Project memasukkan ruang lingkup penilaian</p> <p>TC27 = Leader Project/Member Project memasukkan asumsi penilaian</p> <p>TC28 = Leader Project/Member Project memasukkan sumber penilaian</p> <p>TC29 = Leader Project/Member Project memasukkan model risiko penilaian</p> <p>TC30 = Leader Project/Member Project memasukkan pendekatan analitik penilaian</p> <p>TC31 = Jika data persiapan sudah lengkap, berhasil <i>submit</i> persiapan</p> |
| 8 | <i>Conduct Assessment</i> | <p>TC32 = Leader Project/Member Project memasukan tipe <i>threat source</i></p> <p>TC33 = Leader Project/Member Project memasukan nama <i>threat source</i></p> <p>TC34 = Leader Project/Member Project memasukan deskripsi nama <i>threat source</i></p> <p>TC35 = Leader Project/Member Project memasukan karakteristik <i>threat source</i></p> <p>TC36 = Jika data penilaian sudah lengkap, berhasil submit penilaian</p> |
| 9 | <i>View Risk Assessment Results</i> | TC37 = Leader Project/Member Project melihat hasil penilaian |
| 10 | <i>Setting User</i> | <p>TC38 = Leader Project/Member Project mengubah <i>email</i></p> <p>TC39 = Leader Project/Member Project mengubah <i>username</i></p> <p>TC40 = Leader Project/Member Project mengubah <i>password</i></p> |

Traceability matriks memiliki 6 kolom yang terdiri dari *Use Case* (UC), *Activity Diagram* (AD), *Desain Database* (TB), *Desain Interface* (UI), *Pengujian* (TC), dan verifikasi (Wibowo, 2017). Notasi diambil dari sub bab 3.3. *Template review checklist Traceability* Matriks dapat dilihat pada Lampiran F. Hasil *traceability* matriks dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil *Traceability* Matriks

| No | <i>Use Case</i> | <i>Activity Diagram</i> | Desain <i>Database</i> | Desain <i>Interface</i> | Pengujian | <i>Verified</i> |
|----|-----------------|-------------------------|---------------------------|----------------------------|-------------|-----------------|
| 1 | UC-1 | AD-1 | TB-1 | UI-1 | TC-1 - TC-3 | Ya |
| 2 | UC-2 | AD-2 | TB-1 | UI-2, UI-4, UI-5 | TC-4 - TC-6 | Ya |

| | | | | | | |
|----|-------|-------|------------------------|------------------|---------------|-------|
| 3 | UC-3 | AD-3 | TB-2 | UI-6, UI-7 | TC-7 - TC12 | Ya |
| 4 | UC-4 | AD-4 | TB-1, TB-2, TB-3 | UI-2 | TC-13 - TC-18 | Ya |
| 5 | UC-5 | AD-5 | TB-2 | UI-7, UI-8, UI-9 | TC-19 - TC-21 | Ya |
| 6 | UC-6 | AD-6 | TB-1, TB-2, TB-3 | UI-9 | TC-22 - TC-24 | Ya |
| 7 | UC-7 | AD-7 | TB-2 | UI-10 | TC-25 - TC-31 | Ya |
| 8 | UC-8 | AD-8 | TB-2, TB-4 | UI-11 | TC-32 - TC-36 | Ya |
| 9 | UC-9 | AD-9 | TB-2, TB-4, TB-5, TB-6 | UI-12, UI-13 | TC-37 | Ya |
| 10 | UC-10 | AD-10 | TB-1 | - | TC-38 – TC-40 | Tidak |

Dari Tabel 4.9 dapat dilihat bahwa semua fungsi yang ada di dalam aplikasi berhasil dijalankan dengan baik tanpa ada hambatan. Di bagian perancangan ada 1 bagian yang dilewatkan yaitu desain *interface* untuk fitur *setting user*. Di dalam sistem, fungsi tersebut sudah ada namun tidak digambarkan di bagian desain *interface*.

Metriks untuk *Review Pengujian*

Usaha persiapan, $E_p = 1$ jam

Usaha penilaian, $E_a = 1$ jam

Usaha pengerjaan ulang, $E_r = 1$ jam

$E_{review} = E_p + E_a + E_r = 1 + 1 + 1 = 3$ Jam

Kesalahan total, $Err_{tot} = 1$

Ukuran produk kerja, $WPS = 10$ (berdasarkan total dari *traceability* matriks)

$Error\ Density = \frac{Err_{tot}}{WPS} = \frac{1}{10} = 0,1$

Berdasarkan rumus 2.1 dapat disimpulkan bahwa *review* dengan *traceability* matriks memerlukan waktu selama 3 jam. Hal ini menandakan bahwa waktu melakukan *review* sudah sesuai dengan waktu yang diutamakan. Dengan menggunakan rumus 2.2 ditemukan satu kesalahan di dalam *traceability* matriks. Rumus 2.3 menghasilkan kerapatan kesalahan (*error density*) dalam *traceability* matriks adalah 0,1. Hal ini menunjukkan bahwa kesalahan hanya terjadi kurang dari seperempat keseluruhan matriks.

4.4 Evaluasi Hasil *Technical Review*

Hasil technical review didapatkan setelah melakukan lebih dari 20 pertemuan yang dilakukan oleh peninjau dan pembuat aplikasi *Technical Review* dilakukan mulai dari bagian *use case* diagram. Pada bagian ini ditemukan kerapatan kesalahan sebanyak 0,3 dan waktu pengerjaan selama 3 jam. Berikutnya adalah *activity* diagram. Kerapatan kesalahan sebanyak

0,042 dan waktu pengerjaan selama 3 jam. *Review* selanjutnya adalah desain *database* yang menghasilkan kerapatan kesalahan sebanyak 0 dan waktu pengerjaan selama 3 jam. Tinjauan berikutnya adalah desain *interface* yang memberikan kerapatan kesalahan 0,15 dan waktu pengerjaan 3 jam. *Review* selanjutnya adalah *source code* yang memberikan kerapatan kesalahan 0 dan waktu pengerjaan 5 jam. Tinjauan terakhir adalah *traceability* matriks yang menghasilkan kerapatan kesalahan 0,1 dan waktu pengerjaan 3 jam. Dari data tersebut didapatkan metrik sebagai berikut.

Metriks evaluasi Technical Review

$$E_{\text{review total}} = 3 + 3 + 3 + 3 + 5 + 3 = 20 \text{ Jam}$$

$$\text{Error Density total} = 0,3 + 0,042 + 0 + 0,15 + 0 + 0,1 = 0,592$$

$$\text{Rata-rata } E_{\text{review}} = \frac{20}{6} = 3,33 \text{ jam}$$

$$\text{Rata-rata Error Density} = \frac{0,592}{6} = 0,0987$$

Total waktu tinjauan yang dilakukan adalah 20 jam. Waktu ini terhitung baik karena tidak menghabiskan banyak waktu sehingga waktu pengerjaan aplikasi tidak terlambat karena proses tinjauan. Rata-rata waktu pelaksanaan tinjauan adalah 3,33 jam. Waktu yang didapatkan cukup baik karena waktu rekomendasi melaksanakan *review* adalah 3 jam.

Keseluruhan kerapatan kesalahan adalah 0,592 menandakan bahwa kerapatan kesalahan hanya setengah lebih sedikit saja dari keseluruhan aplikasi Rata-rata kerapatan kesalahan yang terjadi adalah 0,0987. Hal ini membuktikan bahwa kesalahan yang terjadi hanya berupa *error* kecil saja dan tidak berdampak besar terhadap aplikasi.

Berdasarkan testimoni dari pembuat aplikasi, *technical review* kali ini memberikan manfaat terhadap proses pembuatan aplikasi. *Technical review* ini mengungkap beberapa kesalahan yang muncul dan tidak disadari oleh pihak pembuat aplikasi sehingga kesalahan dapat diperbaiki lebih awal. Selain itu juga pertemuan yang dilakukan menghasilkan beberapa ide baru dalam menyelesaikan permasalahan yang muncul ketika proses pembuatan aplikasi. Dengan adanya pertemuan juga memberikan bantuan moril terhadap pembuat aplikasi bahwa ia tidak bekerja sendiri sehingga masalah yang ada dapat didiskusikan bersama.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dihasilkan berdasarkan *technical review* dari aplikasi sistem penilaian risiko bidang keamanan informasi pada sistem informasi menggunakan *National Institute of Standard and Technology* (NIST) SP 800-30 adalah sebagai berikut:

1. Pengujian *technical review* untuk aplikasi ini dilakukan dengan cara melakukan pertemuan antara pembuat aplikasi dan peninjau di satu pertemuan. Pertemuan yang dilakukan sudah direncanakan sebelumnya sehingga pertemuan dapat berjalan secara kondusif
2. Pengujian *technical review* yang dilakukan memberikan manfaat dalam menemukan kesalahan pembuatan dalam aplikasi dan juga dapat menemukan ide baru untuk pemecahan ketika sedang melakukan pertemuan.

5.2 Saran

Saran untuk *technical review* aplikasi lain dengan skala sedang atau besar (dalam artian dikerjakan lebih dari satu orang) adalah pertemuan yang dilakukan harus lebih ditertibkan lagi sehingga hasil tinjauan dapat lebih cepat terlaksana dan kondusif. Selain itu juga dibutuhkan komitmen bersama untuk selalu mengadakan rapat sesuai dengan waktu yang telah ditetapkan. Hal lain yang harus diperhatikan adalah perlu dilakukannya pelatihan untuk setiap peninjau maupun pembuat aplikasi sehingga *technical review* yang dilakukan dapat menemukan dan mengurangi kesalahan lebih banyak. Selain itu validasi *template review checklist* perlu dilakukan untuk penelitian selanjutnya mengenai *technical review*.

DAFTAR PUSTAKA

- A, A. H. (2005). Perencanaan dan Pengendalian Proyek Dengan Metode PERT - CPM : Studi Kasus Fly Over Ahmad Yani, Karawang. *Journal The Winners, Vol. 06 No.2*, 155-174.
- Borysowich, C. (2006, November 26). *ittoolbox*. Diambil kembali dari ittoolbox: <http://it.toolbox.com/blogs/enterprise-solutions/use-case-quality-checklist-20881>
- Chemuturi, M. (2010). *Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers*. Newcastle: J. Ross Publishing.
- Guru99. (2011, Agustus 5). *Guru99*. Diambil kembali dari <https://www.guru99.com/traceability-matrix.html>
- Gutha, S. R. (2015, Agustus 31). *evoketechnologies*. Diambil kembali dari <http://www.evoketechnologies.com/blog/code-review-checklist-perform-effective-code-reviews/>
- Ikhsani, Z. A. (2017). Sistem Penilaian Risiko Bidang Keamanan Informasi pada Sistem Informasi Menggunakan National Institute of Standard and Technology (NIST). *Skripsi*.
- John. (2011, 04 20). *allaboutrequirements*. Diambil kembali dari <http://www.allaboutrequirements.com/2011/04/use-case-completeness-tool-a-checklist.html>
- M. Sidi Mustaqbal, d. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi terapan Volume I, No. 3*.
- NIST. (2012). Guide for Conducting Risk Assessments. *NIST Special Publication 800-30 Revision 1*.
- OWASP. (2008). OWASP Code Review Guide Book V 1.1.
- Pressman, R. S. (2010). *Software Engineering A Practitioner's Approach 7th Edition*. New York: McGraw-Hill.
- Pressman, R. S. (2012). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Shortreed, J. C. (2003). Risk Management Frameworks For Human Health And Environmental Risks. *Journal of Toxicology and Environmental Health. Vol. 6*, 569-720.
- Syafitri, W. (2016). Penilaian Risiko Keamanan Informasi Menggunakan Metode NIST 800-30. *Jurnal CoreIT, Vol.02, No.2*.
- Wibowo, A. F. (2017). Pengujian Perangkat Lunak Berbasis Use Case Menggunakan Traceability Matrix. *Skripsi*.

Widiartanto, Y. H. (2017, April 03). *Kompas*. Diambil kembali dari Kompas:
<http://tekno.kompas.com/read/2017/04/03/11140777/aplikasi.gojek.mendadak.hilang.dari.app.store>

LAMPIRAN

10/3/2017

Laporan Technical Review

Laporan Technical Review

formulir ini di gunakan untuk laporan hasil peninjauan

* Wajib

1. Apa yang ditinjau

2. Siapa yang melakukan Peninjauan

3. Permasalahan apa yang ditemukan dan bagaimana cara-cara untuk mengatasinya *

Diberdasarkan oleh



Google Forms

LAMPIRAN A

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|-------------------------------------------------------------------------------------------------------------------------------------|---------------------|------------|
| 1 | Apakah setiap <i>use case</i> secara tepat menyatakan satu atau lebih kebutuhan fungsional yang sistem harus penuhi? | | |
| 2 | Apakah Setiap <i>use case</i> yang ada sudah sesuai standar? | | |
| 3 | Apakah setiap aktor sudah tepat sesuai aturan penamaan aktor? | | |
| 4 | Apakah kumpulan <i>use case</i> menangkap semua kebutuhan fungsional dari sistem? | | |
| 5 | Apakah <i>use case</i> spesifik sehingga tidak ambigu, dan hanya memiliki satu tafsiran? | | |
| 6 | Apakah kumpulan <i>use case</i> konsisten secara internal, sehingga tidak ada subset <i>use case</i> konflik dengan subset lainnya? | | |
| 7 | Apakah kumpulan <i>use case</i> juga konsisten secara eksternal dengan produk OOAD lainnya? | | |
| 8 | Spesifikasi produk dapat dilakukan pengujian untuk menentukan terpenuhi atau tidaknya kebutuhan dari <i>use case</i> | | |
| 9 | Apakah struktur dan gaya <i>use case</i> mendukung perubahan yang dapat dibuat dengan mudah, lengkap, dan konsisten? | | |
| 10 | Apakah <i>use case</i> mendukung penelusuran ke belakang dan depan? | | |

LAMPIRAN C

| No | Item | Ya/Tidak | Keterangan |
|----|----------------------------------------------------------------------------------------------|----------|------------|
| 1 | Apakah setiap entitas diubah menjadi tabel? | | |
| 2 | Apakah setiap tabel sudah terdapat indeks? | | |
| 3 | Sudahkah setiap <i>foreign key</i> di spesifikasikan? | | |
| 4 | Apakah integritas data sesuai dengan aturan integritas data database management sistem? | | |
| 5 | Apakah privileges dan autoritas, data akses kontrol, dan lain-lain. Sudah diimplementasikan? | | |
| 6 | Apakah nama tabel sudah benar? | | |
| 7 | Apakah nama kolom data sudah benar? | | |
| 8 | Apakah tipe data sudah benar? | | |
| 9 | Apakah penggunaan <i>foreign key</i> sudah tepat? | | |
| 10 | Apakah setiap tabel memiliki <i>primary key</i> ? | | |
| 11 | Apakah penggunaan <i>constraint</i> sudah tepat? | | |
| 12 | Apakah bahasa SQL tidak digunakan sebagai nama objek database? | | |

LAMPIRAN D

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|---------------------------------------------------------------------------------------------------------------------|---------------------|------------|
| 1 | Apakah desain dan teknologi untuk pertukaran informasi setiap komponen, fungsi, dan tugas sudah ditetapkan? | | |
| 2 | Apakah desain dan teknologi dapat bertukar informasi antara setiap <i>interface</i> ? | | |
| 3 | Apakah semua fitur dan kemampuan yang dibutuhkan sudah dibuat? | | |
| 4 | Apakah komponen sistem termasuk perangkat keras, jaringan, sistem operasi, dan database yang dibutuhkan ditetapkan? | | |
| 5 | apakah <i>user interface</i> untuk sistem sudah ditetapkan? | | |
| 6 | Apakah hierarki navigasi layar sudah ditetapkan? | | |
| 7 | Apakah terdapat mockup layar? | | |
| 8 | Apakah informasi atribut yang cocok untuk setiap tampilan sudah di tunjukkan? Apakah keamanan sudah di definisikan? | | |
| 9 | Apakah komponen utama lain dari fungsi pengguna yang membutuhkan interface sudah dapat digunakan? | | |
| 10 | Apakah Desain sudah memiliki bagian <i>error handling</i> ? | | |
| 11 | Apakah ukuran <i>font</i> tulisan sudah cukup jelas? | | |
| 12 | Apakah posisi setiap menu sudah tepat? | | |
| 13 | Apakah tata letak tulisan sudah tepat? | | |

LAMPIRAN E

| No | Item | Hasil (Ya/Tidak) | Keterangan |
|----|----------------------------------------------------------------------------------|---------------------|------------|
| 1 | Apakah setiap <i>method</i> menggunakan bahasa yang jelas? | | |
| 2 | Apakah setiap baris kode yang ditulis bebas dari typo? | | |
| 3 | Apakah setiap kelas, variabel, dan <i>method modifiers</i> sudah tepat? | | |
| 4 | Apakah setiap kebutuhan sistem sudah dibuat ke dalam bentuk <i>source code</i> ? | | |
| 5 | Apakah <i>Source code</i> sudah berukuran sesuai dengan kebutuhan sistem? | | |

LAMPIRAN F

| No | <i>Use Case</i> | <i>Activity Diagram</i> | <i>Desain Database</i> | <i>Desain Interface</i> | <i>Pengujian</i> | <i>Verified</i> |
|----|-----------------|-------------------------|------------------------|-------------------------|------------------|-----------------|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | | | | | | |
| 9 | | | | | | |
| 10 | | | | | | |

LAMPIRAN G

FORM-TA/TF-A3



UNIVERSITAS ISLAM INDONESIA
Jurusan Teknik Informatika FTI

SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Arya

No. Mhs. : _____

Judul TA : _____

Masukkan ke latar belakang mengenai alasan pemilihan
topsis yg dijadikan studi kasus.

Nilai kemajuan Tugas Akhir: _____ (0 - 100)
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 19 Sept 2017

Dosen,

Andhira Giri P.
(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran