

## 1. Pendahuluan

### 1.1 Latar Belakang

*Fraud* didefinisikan sebagai tindakan kecurangan atau penipuan yang disengaja, direncanakan, dan dimaksudkan untuk merampas pihak lain, baik secara langsung maupun tidak langsung [1]. Tindakan ini kerap terjadi karena adanya kesempatan dan tekanan yang mendorong pelaku. Marks [2] berpendapat terdapat lima elemen yang memicu seseorang melakukan penipuan yang disebut *Fraud Pentagon*. Kelima elemen tersebut adalah arogansi (*arrogance*), kompetensi (*competence*), kesempatan (*opportunity*), tekanan (*pressure*), dan rasionalisasi (*rationalization*). Dari lima elemen tersebut dapat diketahui terdapat dua faktor utama alasan seseorang melakukan penipuan, yaitu faktor internal berupa arogansi, kompetensi, dan rasionalisasi, serta faktor eksternal berupa peluang dan tekanan. *Fraud* dapat terjadi dalam banyak kasus, salah satunya kecurangan dalam kasus laporan keuangan. Berdasarkan Survei Fraud Indonesia tahun 2019 yang dilakukan oleh ACFE Indonesia Chapter, kasus penipuan dalam laporan keuangan menempati peringkat ketiga sebagai jenis *fraud* yang paling umum terjadi di Indonesia [3]. Kasus penipuan telah menjadi realitas yang sulit dihindari dalam lingkungan bisnis. Hal ini tentunya menyebabkan kerugian finansial pihak tertentu. Maka dari itu, penting bagi suatu entitas untuk memiliki media yang dapat mencegah dan mendeteksi adanya indikasi *fraud*.

Dalam menghadapi permasalahan *fraud* yang semakin meresahkan, tentunya entitas perlu bertindak cepat untuk menghilangkan tradisi terjadinya *fraud*. Di era digital yang berkembang pesat, penggunaan aplikasi telah menjadi pondasi utama bagi banyak entitas. Oleh karena itu, pengembangan aplikasi *Fraud Deterrence Propeller* (FDP) menjadi suatu langkah proaktif yang krusial untuk mengamankan aset entitas dan mencegah kerugian finansial yang disebabkan oleh tindakan *fraud*. Aplikasi FDP merupakan aplikasi berbasis web yang ditujukan untuk mengatasi permasalahan yang dibahas, yaitu mencegah dan mendeteksi terjadinya *fraud* dalam entitas/perusahaan. Konsep yang diterapkan untuk mencegah *fraud* adalah dengan pengisian *fraud assessment* yang dinamakan *The DETER-E Model*. Lalu, untuk mendeteksi *fraud* menggunakan model analisis *Beneish M-Score* yang melakukan perbandingan data pendapatan dari laporan keuangan tahun yang ditentukan dengan tahun sebelumnya [4]. Karena aplikasi ini berbasis digital, hal ini memungkinkan deteksi *fraud* lebih cepat daripada dilakukan secara manual. Selain itu, sulit untuk melakukan perubahan data atau rekayasa oleh pihak yang tidak memiliki otoritas ke dalam aplikasi. Dengan adanya aplikasi ini, entitas memiliki alat yang dapat memantau dan mendeteksi tanda-tanda kecurangan secara lebih efisien dan efektif, serta memberikan pertahanan tambahan dalam menjaga integritas dan kepercayaan dalam lingkungan bisnis yang semakin kompleks [5].

Untuk memastikan aplikasi yang dikembangkan sesuai dengan kebutuhan pengguna, maka dilakukan identifikasi kebutuhan perangkat lunak tersebut. Hal pertama yang perlu dilakukan dalam pengembangan perangkat lunak atau disebut *Software Development Life Cycle* (SDLC), adalah dengan rekayasa kebutuhan (*requirement engineering*) [6]. Rekayasa kebutuhan merupakan fase penting dalam pengembangan perangkat lunak yang bertujuan untuk mengumpulkan kebutuhan, menganalisis, dan mendokumentasikannya untuk diimplementasikan dalam kode perangkat lunak [7]. Rekayasa kebutuhan yang kurang baik berpotensi menyebabkan kegagalan pada proyek perangkat lunak. Berdasarkan [8], tingkat keberhasilan proyek perangkat lunak hanya 28%. Banyak proyek yang gagal atau tidak selesai sesuai jadwal dikarenakan analisis yang tidak memadai terhadap atribut kebutuhan. Berdasarkan survei yang dilakukan [9], mayoritas *bugs* dalam proyek perangkat lunak terjadi karena kesalahan dalam penulisan kebutuhan. Ini terjadi karena kebutuhan tersebut tidak tertulis, tidak jelas, ambigu, dan salah. Hal ini menjadikan tahap rekayasa kebutuhan merupakan tahap krusial dalam pengembangan perangkat lunak.

Setelah menganalisis kebutuhan pengguna, dilakukan pemodelan spesifikasi yang menggambarkan sistem informasi sesuai kebutuhan pengguna tersebut. Pemodelan tersebut ditujukan untuk menghindari terjadinya kesalahpahaman informasi antara *developer* (pengembang) dengan *project owner* (pemilik proyek). Bahasa pemodelan yang digunakan adalah *Unified Model Language* (UML). Artefak keluaran dari tahap ini adalah dokumen *Software Requirement Specification* (SRS) dan *Software Design Document* (SDD) yang telah divalidasi oleh *stakeholder* [10]. Dirancangnya pemodelan ini bertujuan untuk mempermudah kolaborasi antara tim pengembang, meningkatkan pemahaman bersama, dan membantu mengurangi ambigu dalam spesifikasi dan perancangan sistem aplikasi. Dalam pembuatan SRS atau SDD, seringkali terjadi kesalahan di mana terdapat beberapa kebutuhan pengguna yang terlewat atau tidak diperhitungkan. Hilangnya beberapa kebutuhan menyebabkan ketidakpuasan pengguna akan produk yang tidak lengkap. Kegiatan ini menimbulkan biaya dan waktu tambahan [8]. Maka dari itu, tahap rekayasa kebutuhan merupakan tahap yang krusial dalam pengembangan perangkat lunak.

Rekayasa kebutuhan pada aplikasi *Fraud Deterrence Propeller* (FDP) dilakukan dengan menggunakan pendekatan *Kotonya and Sommerville Linear Process Model*. Siklus dari model ini menunjukkan proses pengembangan linier, mulai dari pengumpulan kebutuhan, analisis kebutuhan, dokumentasi kebutuhan, dan validasi kebutuhan. Model linier ini diusulkan oleh Kotonya dan Sommerville [11]. Alasan dipilihnya model ini karena model bersifat linier yang berarti terletak pada suatu garis lurus, sehingga tahapan yang dilakukan

terstruktur dan terorganisir. Model pendekatan ini juga cocok digunakan untuk proyek berskala kecil atau yang sudah memiliki tujuan jelas [12]. Pengembangan perangkat lunak *Fraud Deterrence Propeller* (FDP) menggunakan metode *Agile*. Karena pengembangan perangkat lunak secara keseluruhan menggunakan metode *Agile*, maka digunakan kerangka kerja *sprint*. Tahap rekayasa kebutuhan ini dilakukan pada *sprint* 1 sampai *sprint* 5.

## 1.2 Topik dan Batasannya

Dari latar belakang yang telah diuraikan sebelumnya, topik yang dipermasalahkan untuk dikerjakan dari penelitian ini adalah bagaimana melakukan perancangan rekayasa kebutuhan pada aplikasi *Fraud Deterrence Propeller* (FDP) dengan menggunakan metode *Kotonya and Sommerville Linear Process Model*?

Terdapat batasan yang ditujukan untuk menentukan ruang lingkup dan fokus penelitian. Dengan adanya batasan ini, diharapkan penelitian dapat lebih terfokus untuk memahami kebutuhan pengguna hingga validasi kebutuhan tersebut. Batasan dari penelitian ini yaitu:

- 1) Pengembangan perangkat lunak dibatasi sampai tahap rekayasa kebutuhan. Adapun, proses rekayasa kebutuhan dalam penelitian ini berfokus pada pengumpulan kebutuhan, analisis kebutuhan, dokumentasi kebutuhan, dan validasi kebutuhan, mengabaikan aspek lain dari rekayasa kebutuhan, seperti manajemen perubahan kebutuhan.
- 2) Dokumen yang dihasilkan dari rekayasa kebutuhan ini adalah dokumen *Software Requirement Specification* (SRS) dan *Software Design Document* (SDD).
- 3) Diagram-diagram yang dihasilkan dalam dokumen SRS meliputi *use case diagram*, *use case description*, *activity diagram*, *sequence diagram*, *class diagram*, dan *Entity Relationship Diagram* (ERD).
- 4) Diagram-diagram yang dihasilkan dalam dokumen SDD adalah *architectural design*, *component diagram* dan *deployment diagram*.

## 1.3 Tujuan

Tujuan yang hendak dicapai dari penelitian ini adalah untuk menganalisis, merancang, dan mengimplementasikan proses rekayasa kebutuhan menggunakan *Kotonya and Sommerville Linear Process Model* pada aplikasi *Fraud Deterrence Propeller* (FDP). Dengan mencapai tujuan ini, penelitian diharapkan dapat memberikan kontribusi dalam mengoptimalkan proses rekayasa kebutuhan dan memastikan keselarasan antara kebutuhan pengguna dengan representasi perangkat lunak yang dikembangkan.

## 1.4 Organisasi Tulisan

Bagian ini menuliskan berbagai bab yang terkandung dalam jurnal Tugas Akhir (TA). Berikut adalah rinciannya:

- 1) Studi Terkait, berisikan studi literatur jurnal yang memiliki keterkaitan dengan topik yang dibahas dalam jurnal TA.
- 2) Sistem yang Dibangun, berisikan tahapan penelitian yang dilakukan.
- 3) Hasil dan Pembahasan, berisikan hasil temuan dari tiap tahapan penelitian yang telah dilakukan.
- 4) Kesimpulan, berisikan simpulan dari seluruh kegiatan penelitian.

## 1.5 Sistematika Pembahasan

Bagian ini menampilkan pihak-pihak yang terlibat dalam proyek pengembangan aplikasi FDP atau disebut *stakeholder*. *Stakeholder* dapat berasal dari dalam ataupun luar organisasi. Untuk menyelesaikan proyek ini, tiap anggota memiliki peran dan tanggung jawab yang telah ditetapkan. Berikut merupakan rincian peran *stakeholder*:

Tabel 1. Pihak yang Terlibat

<b>Peran</b>	<b>Deskripsi</b>
<i>Project Owner</i>	Klien utama atau pemilik proyek yang menjadi acuan dalam proyek ini.
<i>System Analyst</i>	Melakukan pengumpulan kebutuhan pengguna dan menentukan proses bisnis aplikasi yang hendak dikembangkan.
<i>UI/UX Designer</i>	Membuat desain <i>prototype</i> antarmuka dari aplikasi yang hendak dikembangkan.
<i>Front-End Developer</i>	Mengimplementasikan desain <i>prototype</i> antarmuka ke dalam bentuk kode yang menjadi sebuah web untuk diakses.
<i>Back-End Developer</i>	Mengintegrasikan API untuk komunikasi pengguna dari tampilan antarmuka ke basis data.
<i>Quality Assurance 1</i>	Melakukan proses pengujian <i>unit testing</i> dan <i>integration testing</i> pada aplikasi.
<i>Quality Assurance 2</i>	Melakukan proses pengujian terhadap <i>system testing</i> dan <i>performance testing</i> pada aplikasi