

Rekayasa Kebutuhan Teknologi Informasi Tepat Guna Menggunakan Teknik Elisitasi I*Frame dan *Joint Application Design* untuk Aplikasi Cosco pada Proyek Infrastruktur Jalan Tol di Era Industri 4.0

Darmawan Lahru Riatma*, Mashabah

Program Studi D3 Teknik Informatika, Sekolah Vokasi, Universitas Sebelas Maret

*Email: darmawanlr@staff.uns.ac.id

Info Artikel

Kata Kunci :

rekayasa kebutuhan, *elisitasi i*frame*, *joint application design* (JAD)

Kata Kunci :

requirements engineering, rlisitasi I*Frame, joint application design (JAD)

Tanggal Artikel

Dikirim : 23 Oktober 2021

Direvisi : 23 November 2021

Diterima : 30 November 2021

Abstrak

COSCO (*cost control*), merupakan sebuah aplikasi yang di implementasikan untuk membantu mendigitalisasi semua proses kontrol pekerjaan dan pengendalian keuangan jalan tol secara jarak jauh. Dalam pengembangan aplikasi, analisa kebutuhan merupakan tahap penentu sukses atau tidaknya sebuah aplikasi. Maka dari itu, dalam menganalisa kebutuhan COSCO menggunakan metode pendekatan *I*Frame* dan *Joint Application Design* (JAD) agar seluruh kebutuhan dan bisnis proses dari user dapat dipetakan dan terdokumentasi secara spesifik sesuai kebutuhan user. Melalui teknik *I*Frame* peneliti dapat mengidentifikasi sedini mungkin *key stakeholder* yang terlibat dan berpengaruh terhadap keputusan-keputusan pengembangan aplikasi COSCO mulai dari pimpinan hingga staff. Serta melalui metode JAD peneliti dapat mengidentifikasi tujuh dokumen hasil analisa kebutuhan sesuai dengan spesifikasi kebutuhan system. Dokumen tersebut diantaranya adalah Role Peran Pengguna, *Flow Diagram*, Kebutuhan Fungsional, Kebutuhan Non fungsional, *Entity Relationship Diagram*, Tampilan Antar Muka Sistem, Kebutuhan Adaptive Sistem

Abstarct

*COSCO (cost control), is an application that has been implemented to help digitize all processes of work control and financial control of toll roads remotely. In application development, requirements analysis is a critical step in determining the success or failure of an application. Therefore, in analyzing the needs of COSCO using the I*Frame approach and Joint Application Design (JAD) so that all user needs and business processes can be mapped and specifically documented according to user needs. Researchers can use the I*Frame technique to identify key stakeholders, from leadership to staff, who are involved in and influence COSCO application development decisions as early as possible. And through the JAD method, researchers can identify seven documents resulting from the needs analysis following the system requirements specifications. These documents include Role Users, Flow Diagrams, Functional Requirements, Non-Functional Requirements, Entity Relationship Diagrams, User Interface and Adaptive System Requirements.*

1. PENDAHULUAN

Di negara kepulauan yang memiliki 17.508 pulau, sistem jaringan jalan merupakan kebutuhan mendasar untuk menghubungkan masyarakat dan perniagaan dengan pekerjaan, layanan, pasar, mengurangi biaya logistik, dan merangsang pertumbuhan industri di Indonesia. Menjawab kebutuhan tersebut, pemerintah menempatkan konektivitas tinggi sebagai salah satu prioritas utama. Pemerintah membangun dan mengembangkan Jalan Tol Trans-Sumatera. Jalan tol ini akan menghubungkan Lampung dan Aceh melalui 24 ruas jalan berbeda yang panjang keseluruhannya mencapai 2.704 km dan akan beroperasi penuh pada 2024. Pandemi COVID 19 saat ini sangat dinamis dan belum bisa diprediksi kapan akan segera berakhir, sementara proses pekerjaan proyek infrastruktur jalan tol tetap harus dilanjutkan dan selesai sesuai dengan target. Ada keinginan kuat dari stakeholder pengembang proyek untuk investasi di bidang teknologi informasi, demi mendukung proses pekerjaan jalan tol ditengah pandemi. Pandemi COVID 19 ini menuntut semua instansi diseluruh dunia untuk cepat beradaptasi, agar tetap bisa bersaing secara kompetitif dengan kompetitor. Pengembang infrastruktur jalan tol berkeinginan untuk mengembangkan teknologi informasi yaitu aplikasi *cost control*.

Aplikasi COSCO ini di implementasikan untuk membantu mendigitalisasi semua proses kontrol pekerjaan dan pengendalian keuangan jalan tol ditengah pandemi, yang mana semua proses tersebut saat ini masih dilakukan semi manual. Tahapan awal dari pengembangan perangkat lunak adalah analisa kebutuhan, [1] Spesifikasi kebutuhan sebagai bagian dari kebutuhan teknik terutama berurusan dengan bagaimana mengekspresikan kebutuhan secara spesifik, terukur, dapat direalisasikan, dapat dicapai, dan terikat waktu. Pada tahapan ini adalah penentu sukses atau tidaknya aplikasi tersebut, karena menurut Menurut [2], [3] dalam [4] 40% sampai dengan 60% kesalahan dalam suatu proyek pembangunan perangkat lunak yang mungkin muncul pada tahapan berikutnya, berawal dari kesalahan yang dilakukan pada tahap spesifikasi kebutuhan. Hal ini diperkuat dengan [5] berdasarkan survey yang dilakukan oleh standish group, menyatakan bahwa alasan umum kegagalan sebuah proyek bukanlah karena hal teknis tapi terletak pada aspek non teknis yaitu analisis kebutuhan sistem. [3] juga menjelaskan bahwa penyebab tunggal terbesar dari kegagalan pengembangan perangkat lunak tersebut adalah adanya defisiensi pada tahapan spesifikasi kebutuhan. [6] Menemukan bahwa titik-titik defisiensi tersebut tersebar dalam ranah proses, teknologi, dan sumber daya manusianya. Selama spesifikasi persyaratan, para sistem analis fokus pada menentukan persyaratan, yang pada kebanyakan kasus ditulis dalam bahasa alami. Oleh karena itu, persyaratan spesifikasi mewarisi subjektivitas bahasa alami. Ini sering menyebabkan kesalahan umum yang dilakukan oleh sistem analis saat menentukan persyaratan [1]. Kesalahan ini dikenal sebagai tujuh dosa Meyer [1]. [7] Untuk dapat memahami dan menentukan kebutuhan sistem yang baik, seorang analis sistem perlu memahami pengetahuan domain dari sistem. Ada beberapa teknik dalam elisitasi kebutuhan untuk meningkatkan pemahaman pengetahuan domain, seperti wawancara pengguna, kuesioner, analisis dokumen, dan brainstorming.

Berdasarkan survei yang telah dilakukan dalam sebuah penelitian, dijelaskan bahwa sebagian besar pengembang perangkat lunak membuat rekayasa kebutuhan menggunakan bahasa informal [1]. Ini dianggap lebih mudah dipahami dan dikomunikasikan pemangku kepentingan, dengan semakin populernya metode tangkas, pembuatan cerita pengguna berkembang pesat [8]. Penelitian dalam rekayasa kebutuhan telah menerima banyak perhatian di era industri perangkat lunak yang dominan saat ini. Perhatian dalam penyelidikan empiris rekayasa kebutuhan tumbuh secara signifikan secara keseluruhan [12]. Hampir seperlima dari penelitian ini berfokus pada analisis persyaratan, yang meliputi kualitasnya. Mendeteksi kebisingan dalam spesifikasi persyaratan juga telah menarik perhatian dari peneliti rekayasa persyaratan [13] [14] [15]. Penelitian lainnya berfokus pada dampak noise dan cacat pada analisis kebutuhan [16] [17], kebutuhan sistem yang salah [18], spesifikasi yang berlebihan [19] dan ketidaklengkapan.

Pada penelitian ini penulis membantu pengembang proyek jalan tol untuk, menganalisa kebutuhan perangkat lunak menggunakan metode pendekatan *I*Frame* dan *Joint Application Design*. Menurut [6] Kerangka *I*Frame* membantu sistem analis menemukan atau mengidentifikasi para pemangku kepentingan yang terlibat dari sistem tersebut. Selain itu, kerangka ini juga memfasilitasi suatu perbandingan atas konfigurasi-konfigurasi berbeda sedini mungkin. Kerangka *I*Frame* ini akan di kombinasikan dengan metode *Joint Application Design*. [8] *Joint Application Development* (biasa dikenal sebagai JAD) adalah sebuah teknik pengumpulan data yang mengizinkan anggota tim proyek, pengguna, dan manajemen untuk bekerja sama untuk mengidentifikasi persyaratan untuk sistem. JAD diklaim sebagai salah satu metode yang paling berguna untuk mengumpulkan informasi dari pengguna dan dapat mengurangi *scope creep* hingga 50%. JAD mengembangkan suatu proses yang memfasilitasi pembentukan visi tentang "seharusnya sistem seperti apa". Melalui JAD, pengembang membantu pengguna merumuskan masalah dan mengeksplorasi solusi, serta menambahkan rasa keterlibatan pengguna, kepemilikan dan komitmen untuk

keberhasilan sistem. Harapan peneliti dengan menggunakan kerangka *I*Frame* dan metode *Joint Application Design* ini seluruh kebutuhan aplikasi COSCO dapat teridentifikasi dan didokumentasikan dengan baik, akurat serta terukur, sehingga nantinya pada tahapan pengembangan sistem hasilnya sesuai dengan kebutuhan bisnis proses dari user.

2. METODE PENELITIAN

Ada empat tahapan Metodologi penelitian untuk menujung penelitian ini, tahapan tersebut adalah menjalin kerja sama dengan mitra, mempelajari bisnis proses pengembang jalan tol, elisitasi kebutuhan perangkat lunak dan hasil dokumen spesifikasi perangkat lunak. Berikut ini adalah detail tahapan yang digunakan:

2.1 Menjalinkan Komunikasi Dengan Mitra

Tahapan menjalin kerjasama dengan pengembang proyek sudah dilakukan oleh peneliti 1, komunikasi yang berjalan dengan baik ditandai dengan adanya kerja sama antara kami sebagai peneliti dengan pengembang proyek. Stakeholder pengembang proyek sangat antusias untuk memulai kerja sama, karena adanya kebutuhan yang mendesak untuk segera diselesaikan guna menjaga stabilitas progres pekerjaan ditengah pandemi seperti saat ini.

2.2 Domain Understanding Mitra

Tahapan kedua adalah pengenalan lingkungan internal pengembang proyek, dalam tahapan ini peneliti mempelajari dan memahami lingkungan internal pengembang. Komunikasi awal dengan stakeholder sangat penting dilakukan sebagai langkah awal untuk persiapan elisitasi kebutuhan. Komunikasi kami lakukan dengan stakeholder pengembang mulai jajaran direksi, kepala departemen, staff pengendalian biaya dan kepala proyek. Pada tahapan ini peneliti 1 dan peneliti 2 berkontribusi langsung dalam menjalin komunikasi awal.

2.3 Elisitasi Kebutuhan Perangkat Lunak

Setelah informasi awal didapat tahapan selanjutnya adalah elisitasi kebutuhan user. Pada penelitian ini proses elisitasi menggunakan teknik *I*Frame* dan Join Desain Application (JAD). Pendekatan *I*Frame* digunakan membantu peneliti menemukan atau mengidentifikasi para pemangku kepentingan yang terlibat dari sistem tersebut. JAD pada penelitian ini digunakan sebagai alat bantu dalam tahapan elisitasi kebutuhan yang berfungsi untuk mengembangkan kerja sama, pemahaman dan kerja tim antara pembeli, pengguna, dan peneliti satu serta peneliti dua.

2.4 Dokumen Spesifikasi Kebutuhan Perangkat Lunak

Hasil akhir elisitasi kebutuhan ini adalah dokumen spesifikasi kebutuhan perangkat lunak yang terdiri dari dokumen spesifikasi dan dokumen klasifikasi perangkat lunak. Diharapkan dengan teknik elisitasi tersebut dapat menghasilkan dokumen level 3 sampai level 4 yaitu lengkap dan sangat lengkap.

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan kerangka *I*Frame* dan metode *Joint Application Design* (JAD) untuk mendukung proses analisa kebutuhan sistem COSCO. Berikut ini adalah detail hasil pembahasan menggunakan kerangka *I*Frame* dan JAD.

3.1. Hasil Dan Pembahasan Kerangka *I*Frame*

Kerangka *I*Frame* digunakan untuk mengidentifikasi para pemangku kepentingan yang terlibat menggunakan sistem tersebut. Langkah pertama diawali dengan membuat forum grup diskusi (FGD) dengan melibatkan *stakeholder* pengembang proyek dan peneliti. Peneliti sebagai katalisator antara sistem yang akan dikembangkan dengan user yang akan menggunakan sistem tersebut. Kegiatan FGD dilakukan guna mengidentifikasi masukan serta ide dari semua *stakeholder* mitra, mulai dari pimpinan hingga staff. Hasil identifikasi pemangku kepentingan yang terlibat menggunakan sistem tersaji dalam Tabel 1.

Tabel 1. Stakeholder yang terlibat dalam pengembangan sistem

Nama	Jabatan	Internal/ eksternal	Projek Role
Maya	Wakil Direktur Keuangan	Internal	Pemberi dana
Ahmad	Kepada Departemen Pengendalian Keuangan	Internal	Penanggung jawab
Yanuar	Manajer Pengendalian Waktu Proyek	Internal	Senior manager
Muhtarom	Manager pengendalian Sumberdaya Proyek	Internal	Senior manager
Saepuluh	Staff Pengendali Keuangan	Internal	Kepala proyek
Samsul	Staff Pengendalian Waktu	Internal	User

Pada penelitian ini kerangka *I*Frame* sangat membantu peneliti dalam mengidentifikasi sedini mungkin siapa *key stakeholder* yang terlibat dan berpengaruh terhadap keputusan-keputusan pengembangan aplikasi COSCO. Identifikasi *stakeholder* penting dilakukan di awal karena hal ini menentukan keberhasilan proses analisa kebutuhan sistem. Hasil penelitian menunjukkan semua *stakeholder* yang terlibat menggunakan sistem sudah teridentifikasi sejak kegiatan FGD pertama kali dilakukan dengan mitra. Selanjutnya peneliti aktif berkomunikasi dengan *stakeholder* untuk keberhasilan pengembangan aplikasi COSCO.

3.2. Hasil Dan Pembahasan JAD

Penggalan kebutuhan sistem dari *stakeholder* pengguna ini menggunakan metode JAD, melalui JAD peneliti membantu pengguna dalam merumuskan masalah dan mengeksplorasi solusi serta menambahkan rasa keterlibatan pengguna untuk keberhasilan sistem. Hasil analisa kebutuhan sistem menggunakan JAD menghasilkan suatu dokumen pengembangan sistem yang sangat detail dan terukur sesuai dengan kebutuhan *stakeholder* pengguna. Jenis dokumen yang dihasilkan setelah menggunakan metode elisitasi JAD tersaji dalam Tabel 2.

Tabel 2. Dokumen hasil analisis menggunakan metode elisitasi JAD

No	Jenis Dokumen	Status Dokumen
1	Role Peran Pengguna	Sangat Lengkap Level 4
2	Flow Diagram	Sangat Lengkap Level 4
3	Kebutuhan Fungsional	Sangat Lengkap Level 4
4	Kebutuhan Non fungsional	Sangat Lengkap Level 4
5	Entity Relationship Diagram	Sangat Lengkap Level 4
6	Tampilan Antar Muka Sistem	Sangat Lengkap Level 4
7	Kebutuhan Adaptive Sistem	Lengkap Level 3

Jenis dokumen yang dihasilkan adalah Role Peran Pengguna, *Flow Diagram*, Kebutuhan Fungsional, Kebutuhan Non fungsional, *Entity Relationship Diagram*, Tampilan Antar Muka Sistem, Kebutuhan Adaptive Sistem. Kolom status dokumen menunjukkan tingkat status dokumen, level 1 berarti data kebutuhan sistem yang dihasilkan tidak lengkap, level 2 berarti data kebutuhan sistem yang dihasilkan kurang lengkap, level 3 berarti data kebutuhan sistem yang dihasilkan lengkap, dan level 4 berarti data kebutuhan sistem yang dihasilkan sangat lengkap.

Tabel 2 menghasilkan data dan dokumen kebutuhan sistem pada enam jenis dokumen menghasilkan data yang sangat lengkap, sedangkan ada satu jenis dokumen yang menghasilkan data lengkap di level 3 yaitu kebutuhan adaptive sistem.

4. KESIMPULAN

Penerapan kerangka **Frame* berjalan dengan baik, dan mampu mengidentifikasi pemangku kepentingan mitra yang nantinya terlibat langsung dalam menggunakan aplikasi COSCO. Metode *Joint Application Design* (JAD), membantu pengguna dalam merumuskan masalah dan mengeksplorasi solusi serta menambahkan rasa keterlibatan pengguna untuk keberhasilan aplikasi COSCO yang sedang dikembangkan, sehingga dapat dihasilkan enam jenis dokumen kebutuhan sistem dengan status sangat lengkap level 4, sedangkan ada satu jenis dokumen yang menghasilkan data lengkap di level 3.

DAFTAR PUSTAKA

- [1] Siahaan, Ummami (2011), "Natural Language Processing For Detecting Forward Reference In A Dokumen" IPTEK, The Journal for Technology and Science, Vol. 22.
- [2] Davis, A M. (1993). *Software Requirements: Object, Functions, and States.*: Englewood Cliffs, NJ: Prentice Hall PTR.
- [3] Leffingwell, Dean. (1997). "Calculating the Return on Investment from More Effective Requirements Management." *American Programmer* 10(4):13-16.
- [4] Siahaan Daniel (2012), *Analisa Kebutuhan Rekayasa Perangkat Lunak*, Andi Yogyakarta.
- [5] Rosyadi, I. J., 2013. *Analisa dan Perancangan Sistem Informasi Persediaan Barang dengan menggunakan metode FAST pada CV. Tri Jaya*. S1. Universitas Brawijaya.
- [6] Hofmann, H.F. and Lehner, F. (2001). *Requirements Engineering as a Success Factor in Software Projects*. IEEE Software.
- [7] Indra, Daniel, Chastine (2019), "User Story Extraction From Online News For Software Requirement Elicitation : A Conceptual Model". 16th International Joint Conference On Computer Science And Software Engineering (JCSSE).
- [8] Yu E. (1997), "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", *Proceedings of the 3rd IEEE Int. Symp. On Requirements Engineering (RE'97)*, Washington D.C., USA.
- [9] Kassab M. The changing landscape of requirements engineering practices over the past decade. In: 5th International Workshop on Empirical Requirements Engineering, EmpiRE 2015 Institute of Electrical and Electronics Engineers Inc.; 2016. p. 1–8.
- [10] Bakar NH, Kasirun ZM, Salleh N. Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review. *Journal of Systems and Software* 2015 8;106:132–149.
- [11] Alan Dennis, Barbara Haley Wixom, Roberta M Roth. (2013). *System Analysis and Design*. 05. Wiley. Singapore.
- [12] Ambreen, T.; Ikram, N.; Usman, M.; Niazi, M. Empirical research in requirements engineering: Trends and opportunities. *Requir. Eng.* 2018, 23, 63–95. <https://doi.org/10.1007/s00766-016-0258-2>.
- [13] Manek, P.G.; Siahaan, D. Noise Detection In Software Requirements Specification Document Using Spectral Clustering. *JUTI J. Ilm. Teknol. Inf.* 2019, 17, 30–37. <https://doi.org/10.12962/j24068535.v17i1.a771>.
- [14] Fahmi, A.A.; Siahaan, D.O. Algorithms Comparison For Non- Requirements Classification Using The Semantic Feature of Software. *IPTEK J. Technol. Sci.* 2020, 31, 1–9. <https://doi.org/10.12962/j20882033.v31i3.7606>.
- [15] Romano, S.; Scanniello, G.; Fucci, D.; Juristo, N.; Turhan, B. Poster: The effect of noise on requirements comprehension. In *Proceedings of the International Conference on Software Engineering, Madrid, Spain, 23–29 September 2018*; IEEE Computer Society: Los Alamitos, CA, USA, 2018, pp. 308–309. <https://doi.org/10.1145/3183440.3194984>.
- [16] Jørgensen, M.; Grimstad, S. The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. *IEEE Trans. Softw. Eng.* 2011, 37, 695–707. <https://doi.org/10.1109/TSE.2010.78>.

- [17] Ferrari, A.; Gori, G.; Rosadini, B.; Trotta, I.; Bacherini, S.; Fantechi, A.; Gnesi, S. Detecting requirements defects with NLP patterns: An industrial experience in the railway domain. *Empir. Softw. Eng.* 2018, 23, 3684–3733. <https://doi.org/10.1007/s10664-018-9596-7>.
- [18] Alshazly, A.A.; Elfatary, A.M.; Abougabal, M.S. Detecting defects in software requirements specification. *Alex. Eng. J.* 2014, 53, 513–527. <https://doi.org/10.1016/j.aej.2014.06.001>.
- [19] Chari, K.; Agrawal, M. Impact of incorrect and new requirements on waterfall software project outcomes. *Empir. Softw. Eng.* 2018, 23, 165–185. <https://doi.org/10.1007/s10664-017-9506-4>.
- [20] Martínez, A.B.B.; Arias, J.J.P.; Vilas, A.F. Merging requirements views with incompleteness and inconsistency. In *Proceedings of the Australian Software Engineering Conference, ASWEC, Brisbane, Australia, 29 March–1 April 2005*; pp. 58–67. <https://doi.org/10.1109/aswec.2005.29>.