

PERBAIKAN *METHODOLOGY FOR REQUIREMENTS ENGINEERING TECHNIQUE SOLUTION (MRETS)* DENGAN MENAMBAHKAN *DELIVERABLES* KE DALAM ATRIBUT PROYEK DAN TEKNIK *SINGLE SOLUTION*

*** Firli Irhamni, ** Daniel O Siahaan**

* Jurusan Teknik Informatika, Universitas Trunojoyo

Jl. Raya Telang PO. BOX 2, Kamal, Bangkalan, Madura, 69162

** Jurusan Teknik Informatika, ITS

Jl. Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111

E-Mail: * firli@trunojoyo.ac.id, ** daniel@if.its.ac.id

Abstract

The problem which often occurs in developing software engineering is the adding cost from the budget, lateness, and the quality which is in appropriate with the needs of software users. One of the reasons is the unwell the process of engineering. The usage of the needs engineering technique in developing software project is potential to determine the success of developing software project. One of the researches of the matters is combining the needs engineering technique by Li Jiang. The research explained how to select the combination of needs engineering technique using Methodology for Requirements Engineering Technique Solution (MRETS). The result of MRETS method is used to support analys system to decide which of technique will be used. This research is the improvement of MRETS method by adding deliverables and alogiritm rule and applying single solution. The result of this research is needs engineering technique which is suitable with the process of needs engineering. This research contributes to make efficient the use of needs engineering technique.

Kata Kunci: MRETS, deliverables, single solution.

PENDAHULUAN

Penggunaan teknik rekayasa kebutuhan yang sesuai dalam membangun proyek perangkat lunak berpotensi menentukan kesuksesan sebuah proyek rekayasa perangkat lunak. Hasil *report* dari *Standish Group* [1] menjelaskan bahwa penggunaan rekayasa kebutuhan yang tepat memberikan kontribusi terhadap keberhasilan sebuah proyek perangkat lunak.

Beberapa tahun terakhir, telah banyak peneliti dan praktisi yang melakukan penelitian tentang pemilihan teknik rekayasa kebutuhan dan model dalam pengembangan proyek rekayasa perangkat lunak. *Alan* [2] menyatakan bahwa penerapan teknik rekayasa kebutuhan yang sesuai dengan kebutuhan masalah adalah penting untuk menganalisa kebutuhan yang efektif. Hal ini dikuatkan dengan penelitian *Glass* [3] yang menyatakan

bahwa perlunya sebuah metodologi untuk mengembangkan proyek perangkat lunak yang sesuai. Sampai saat ini telah banyak penelitian yang dilakukan tentang pentingnya teknik rekayasa kebutuhan dalam mendukung proses membuat model, dokumen, verifikasi, dan validasi kebutuhan.

Salah satu penelitian mengenai teknik rekayasa kebutuhan dalam pengembangan rekayasa perangkat lunak adalah melakukan seleksi kombinasi teknik rekayasa kebutuhan yang dilakukan *Li Jiang* [4]. Dalam penelitian tersebut dijelaskan seleksi kombinasi dari teknik rekayasa kebutuhan dengan menggunakan metodologi seleksi dan di akhir penelitian dilakukan studi kasus pada proses rekayasa perangkat lunak dalam industri rekayasa perangkat lunak.

Penelitian *Li Jiang* menggunakan model proses rekayasa kebutuhan yang dikembangkan

oleh Kotonya dan Sommerville [5] yang berisi empat tahapan yaitu: *requirements elicitation, requirements analysis and negotiation, requirements documentation, dan requirements verification and validation*. Penelitian ini juga menghasilkan sebuah metodologi dalam melakukan seleksi kombinasi teknik rekayasa kebutuhan yang dinamakan *Methodology for Requirements Engineering Techniques Selection (MRETS)*. Metode *MRETS* menghasilkan beberapa teknik rekayasa kebutuhan yang akan menjadi bahan pertimbangan bagi sistem analis maupun programmer dalam menentukan teknik rekayasa kebutuhan yang akan digunakan.

Penelitian ini melakukan pengembangan terhadap metode *MRETS* dalam dua hal, yaitu:

1. Penambahan atribut *deliverables* dalam masukan dari metode *MRETS*, dan
2. Penerapan *single solution* pada seleksi kombinasi teknik rekayasa kebutuhan yang berbasis metode *MRETS*.

Hasil dari penelitian ini adalah kombinasi teknik rekayasa kebutuhan sesuai dengan proses rekayasa kebutuhan. Pengembangan metode *MRETS* secara umum dapat dilihat pada Gambar 1.

Seleksi Kombinasi Teknik Rekayasa Kebutuhan

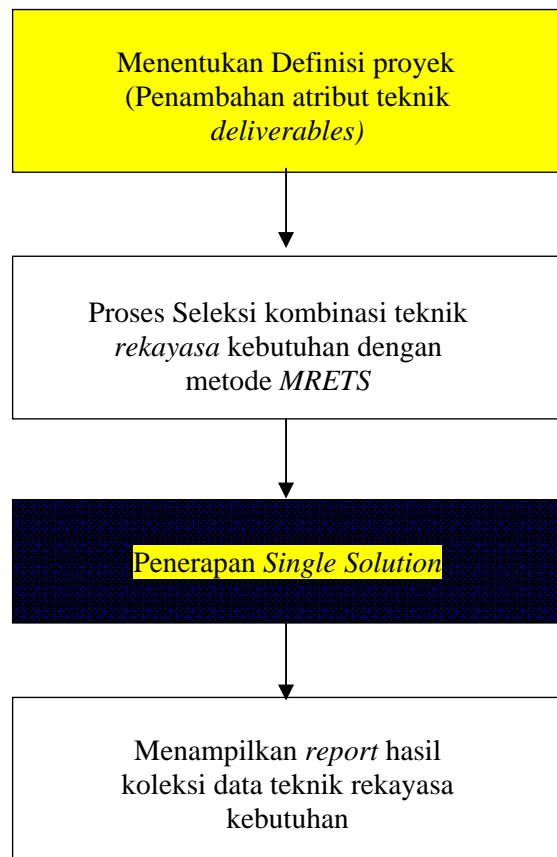
Beberapa tahun terakhir telah banyak dikembangkan teknik rekayasa kebutuhan. Sehingga dibutuhkan kombinasi teknik rekayasa kebutuhan dalam penggunaannya. Beberapa hal yang menjadi alasan dibutuhkannya kombinasi teknik rekayasa kebutuhan:

1. Perbedaan *domain* permasalahan membutuhkan teknik yang berbeda. Sebagai contoh, kebutuhan dari sistem *real-time* membutuhkan analisa dan verifikasi harus tepat. Dalam hal ini dibutuhkan metode yang formal dan tetap untuk proyek perangkat lunak tipe ini.
2. Proyek perangkat lunak yang berbeda membutuhkan teknik yang berbeda. Beberapa proyek dibutuhkan sangat cepat sehingga tidak memperlumahkan kualitas dari produk perangkat lunak tersebut. Tetapi ada juga proyek perangkat lunak yang mempunyai tingkat kesulitan sangat tinggi

sehingga membutuhkan waktu lama. Untuk kasus ini, teknik *Ethnography* sangat baik untuk digunakan.

3. Banyaknya keterlibatan *stakeholder* membutuhkan perbedaan teknik yang digunakan. Beberapa proyek membutuhkan keterlibatan bermacam-macam *stakeholder*. Memaksimalkan keterlibatan *stakeholder* sangat diperlukan dalam kasus ini. Teknik *Focus Group* sangat membantu dalam kasus ini.
4. Sampai saat ini belum ada satu teknik yang dapat menyelesaikan semua permasalahan proses rekayasa kebutuhan. Hal ini berarti kebutuhan *engineers* harus tepat dalam memilih dan mengkombinasikan teknik yang dipakai dalam sebuah proyek perangkat lunak.

Sebelum melakukan seleksi kombinasi teknik rekayasa kebutuhan dibutuhkan terlebih dulu informasi mengenai proyek perangkat lunak tersebut.



Gambar 1. Alur proses pengembangan metode *MRETS*.

Tabel 1. Derajat Nilai Atribut Proyek terhadap Teknik Rekayasa Kebutuhan [4].

| No. | Atribut Proyek | | Teknik <i>Requirements Engineering</i> | | | | | <i>Interview</i> |
|-----------------|-------------------------------------|---|--|-------------------------------|------------------------|--------------------|--------------------|------------------|
| | Nama Atribut | Nilai | <i>Brain Storming</i> | <i>Designer As Apprentice</i> | <i>Document Mining</i> | <i>Ethnography</i> | <i>Focus Group</i> | |
| 1 | <i>Project Size</i> | <i>Very Big</i> | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1 |
| | | <i>Big</i> | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |
| | | <i>Medium</i> | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 1 |
| | | <i>Small</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Very Small</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | <i>Project Complexity</i> | <i>Very High</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>High</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Medium</i> | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 1 |
| | | <i>Low</i> | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |
| | | <i>Very Low</i> | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1 |
| 3 | <i>Requirements Volatility</i> | <i>Very High</i> | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1 |
| | | <i>High</i> | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |
| | | <i>Medium</i> | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 1 |
| | | <i>Low</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Very Low</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | <i>Project Category</i> | <i>Organic Category</i> | 1 | 0.5 | 1 | 0.5 | 1 | 1 |
| | | <i>Semi-detached Category</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Embedded Hardware/ Software System</i> | 1 | 1 | 0.5 | 1 | 0.5 | 1 |
| | | <i>Communication related</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Degree of Safety Criticality</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | <i>Degree of Safety Criticality</i> | <i>High</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Medium</i> | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 1 |
| | | <i>Low</i> | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |
| | | <i>Very Low</i> | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1 |
| | | 6 | <i>Time Constraints</i> | <i>Very High</i> | 0.5 | 0 | 0.25 | 0 |
| <i>High</i> | 0.75 | | | 0.25 | 0.5 | 0.25 | 0.5 | 1 |
| <i>Medium</i> | 1 | | | 0.5 | 0.75 | 0.5 | 0.75 | 1 |
| <i>Low</i> | 1 | | | 0.75 | 1 | 0.75 | 1 | 1 |
| <i>Very Low</i> | 1 | | | 1 | 1 | 1 | 1 | 1 |
| 7 | <i>Cost Constraints</i> | <i>Very High</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>High</i> | 1 | 1 | 1 | 1 | 1 | 1 |
| | | <i>Medium</i> | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 1 |
| | | <i>Low</i> | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |
| | | <i>Very Low</i> | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.5 |

Tabel 2. Teknik Rekayasa Kebutuhan.

| No. | Teknik <i>Requirements Engineering</i> | Proses <i>Requirements Engineering</i> |
|-----|--|---|
| 1 | <i>Brain Storming</i> | <i>Elicitation</i> |
| 2 | <i>Designer As Apprentice</i> | <i>Elicitation</i> |
| 3 | <i>Document Mining</i> | <i>Elicitation</i> |
| 4 | <i>Ethnography</i> | <i>Elicitation</i> |
| 5 | <i>Focus Group</i> | <i>Elicitation</i> |
| 6 | <i>Interview</i> | <i>Elicitation</i> |
| 7 | <i>Contextual Inquiry</i> | <i>Elicitation</i> |
| 8 | <i>Laddering</i> | <i>Elicitation</i> |
| 9 | <i>Viewpoints-Oriented Elicitation</i> | <i>Elicitation</i> |
| 10 | <i>Exploratory Prototype</i> | <i>Elicitation, Analysis & Negotiation, Verification & Validation</i> |
| 11 | <i>Evolutionary Prototypes</i> | <i>Elicitation, Analysis & Negotiation, Verification &</i> |

| | |
|---|--|
| | <i>Validation</i> |
| 12 <i>Viewpoints-Oriented Analysis</i> | <i>Analysis & Negotiation</i> |
| 13 <i>Repertory Grids</i> | <i>Elicitation</i> |
| 14 <i>Scenario Approach</i> | <i>Elicitation, Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 15 <i>Joint Application Development (JAD)</i> | <i>Elicitation</i> |
| 16 <i>Soft Systems Methodology (SSM)</i> | <i>Elicitation</i> |
| 17 <i>Goal-Oriented Analysis</i> | <i>Analysis & Negotiation</i> |
| 18 <i>Viewpoints-Oriented Documentation</i> | <i>Documentation</i> |
| 19 <i>Future Workshops</i> | <i>Elicitation</i> |
| 20 <i>Representation Modeling</i> | <i>Elicitation, Analysis & Negotiation</i> |
| 21 <i>Functional Decomposition</i> | <i>Analysis & Negotiation</i> |
| 22 <i>Decision Tables</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 23 <i>State Machine</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 24 <i>State Charts</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 25 <i>Petri-Nets</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 26 <i>Structured Analysis (SA)</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 27 <i>Real-Time Structure Analysis</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 28 <i>Object-Oriented Analysis</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 29 <i>Problem Frame Oriented Analysis</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 30 <i>Goal-Oriented Verification and Validation</i> | <i>Verification & Validation</i> |
| 31 <i>Entity Relationship Diagrams</i> | <i>Documentation</i> |
| 32 <i>AHP</i> | <i>Analysis & Negotiation</i> |
| 33 <i>Card Shorting</i> | <i>Elicitation</i> |
| 34 <i>Software QFD</i> | <i>Elicitation, Analysis & Negotiation</i> |
| 35 <i>Fault Tree Analysis</i> | <i>Elicitation, Analysis & Negotiation</i> |
| 36 <i>Structured Natural Language Spesification</i> | <i>Documentation</i> |
| 37 <i>Viewpoints-Oriented Verification and Validation</i> | <i>Verification & Validation</i> |
| 38 <i>Unified Modelling Language (UML)</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 39 <i>Z</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 40 <i>LOTOS</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 41 <i>SDL</i> | <i>Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 42 <i>XP</i> | <i>Elicitation, Analysis & Negotiation, Documentation, Verification & Validation</i> |
| 43 <i>Formal Requirements Inspection</i> | <i>Verification & Validation</i> |
| 44 <i>Requirements Testing</i> | <i>Verification & Validation</i> |
| 45 <i>Requirements Checklists</i> | <i>Verification & Validation</i> |
| 46 <i>Utility Test</i> | <i>Verification & Validation</i> |

Atribut Proyek

Atribut proyek adalah deskripsi karakter yang berhubungan dengan proyek perangkat lunak tersebut. Terdapat tujuh atribut proyek dalam

penelitian ini: *Project Size*, *Project Complexity*, *Kebutuhan Volatility*, *Project Category*, *Degree of Safety Critically*, *Time Constraint*, dan *Cost Constraint*. Setiap atribut proyek dari tujuh atribut proyek tersebut mempunyai beberapa pilihan masukan. Misalnya atribut *Project Size* mempunyai alternatif pilihan: *Very Big* (untuk kebutuhan-kebutuhan lebih besar sama dengan 4000), *Big* (untuk kebutuhan-kebutuhan antara 1000 sampai 4000), *Medium* (untuk kebutuhan-kebutuhan 500 sampai 1000), *Small* (untuk kebutuhan-kebutuhan 100 sampai 500), dan *Very Small* (untuk kebutuhan-kebutuhan kurang dari 100). Setiap atribut mempunyai derajat nilai atribut proyek terhadap teknik rekayasa kebutuhan mengambil dari data statistik penelitian sebelumnya [4]. Beberapa atribut proyek dapat dilihat pada Tabel 1.

Teknik dan Proses Rekayasa Kebutuhan

Sampai saat ini sudah dikembangkan sekitar 46 teknik rekayasa kebutuhan. *Kotonya and Sommerville* [5] mengenalkan proses rekayasa kebutuhan yaitu:

1. *Requirements Elicitation*

Merupakan proses identifikasi kebutuhan dan sebagai penghubung antara komunitas yang dilibatkan untuk menemukan dan menyaring kebutuhan berdasarkan batasan dari komunitas tersebut.

2. *Requirements Analysis and Negotiation*

Merupakan proses sebelum kebutuhan tersebut dianalisa dan dimodelkan, dan apabila terjadi permasalahan dapat diselesaikan diantara *stakeholders* terlebih dahulu. Masukan dari *Requirements Analysis and Negotiation* adalah hasil proses *Elicitation*, sedangkan keluaran dari proses ini adalah data kebutuhan yang sudah lengkap dan konsisten.

3. *Requirements Documentation*

Merupakan proses dokumentasi terhadap kebutuhan yang telah disepakati pada suatu tingkatan secara detil dalam bentuk notasi berdasarkan struktur dokumen yang baik. *Requirements Documentation* menerima masukan dari *Requirements Analysis and Negotiation*. Sedangkan keluaran dari proses ini adalah struktur dokumen yang baik dan sudah dispesifikasikan. Dokumen

Tabel 3. Derajat Nilai Atribut Proyek *Deliverables* terhadap Teknik Rekayasa Kebutuhan.

ini kemudian telah siap untuk dilakukan verifikasi dan validasi.

4. *Requirements Verification and Validation*

Merupakan proses menguji dokumen kebutuhan untuk memastikan bahwa dokumen kebutuhan tersebut tidak membingungkan, konsisten, dan lengkap. Proses ini juga membantu *stakeholders* untuk mencapai kepuasan terhadap spesifikasi kebutuhan yang telah diolah. Hal penting yang perlu diperhatikan adalah *Requirements Verification and Validation* tidak mengulang tugas dari *Requirements Analysis and Negotiation* melainkan menggunakan sistem yang akan dibangun sebagai dasar verifikasi dan validasi. Keluaran dari proses ini adalah hasil akhir dokumen spesifikasi kebutuhan yang telah disepakati dan mendapat persetujuan dari semua *stakeholders*.

5. *Requirements Management*

Merupakan proses melakukan identifikasi, organisasi, dokumentasi, dan penelusuran perubahan kebutuhan yang akan mengakibatkan perubahan spesifikasi kebutuhan.

Keseluruhan teknik rekayasa kebutuhan dapat dilihat pada Tabel 2.

Penambahan Atribut Proyek *Deliverables*

Atribut proyek *deliverables* ditambahkan dalam atribut proyek tersebut. Untuk derajat nilai pada tabel atribut proyek adalah bernilai satu apabila atribut proyek *deliverables* tersebut dihasilkan oleh teknik rekayasa kebutuhan dan bernilai nol apabila tidak dihasilkan teknik rekayasa kebutuhan.

Misalnya teknik *Object Oriented Analysis* menghasilkan atribut proyek *deliverables diagram* dan dokumentasi *Uses Case* maka atribut proyek tersebut bernilai satu terhadap teknik *Object Oriented Analysis*. *Engineers* dan *stakeholder* yang terkait dalam proyek perangkat lunak dapat memilih atribut proyek *deliverables* lebih dari satu pilihan.

Data derajat nilai atribut proyek *deliverables* terhadap teknik rekayasa kebutuhan terdapat pada Tabel 3.

| No. | Nama Atribut | Nilai | Brain Storming | Designer As Apprentice | Document Mining | Ethno graphy | Focus Group | Interview |
|-----|---------------------|--|----------------|------------------------|-----------------|--------------|-------------|-----------|
| 8 | <i>Deliverables</i> | Data kebutuhan sistem | 1 | 0 | 0 | 0 | 0 | 0 |
| | | Data <i>survey</i> dan <i>questioner</i> | 0 | 0 | 1 | 0 | 0 | 0 |
| | | Data keadaan lingkungan dan sosial dari sistem | 0 | 0 | 0 | 1 | 0 | 0 |
| | | Data user (struktur organisasi, laporan perusahaan, deskripsi kerja, kebijakan aturan, sistem kerja saat ini). | 0 | 1 | 0 | 0 | 1 | 0 |
| | | Data <i>Interview user</i> | 0 | 0 | 0 | 0 | 0 | 1 |
| | | Data permasalahan sistem saat ini | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Data <i>Interview expert</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Data kebutuhan sistem menurut <i>expert</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa koleksi data & <i>requirements</i> untuk sistem yang baru | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa berdasarkan sudut pandang seluruh bagian <i>user</i> yang terlibat | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Dokumen analisa skenario | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Diagram seluruh bagian dari sistem (permasalahan & pengembangannya) | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Data <i>Use Case & Actor</i> (Desain & Implementasi) | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa <i>decision tables</i> dan <i>rules</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa dalam bentuk <i>State Transition Diagram (STD)</i> untuk 1 <i>state</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa dalam bentuk <i>State Transition Diagram (STD)</i> untuk <i>state</i> lebih dari 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa dalam bentuk diagram <i>State Transition Petri-Net</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa data <i>flow</i> dan proses terstruktur | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa permasalahan sistem | 0 | 0 | 0 | 0 | 0 | 0 |
| | | <i>ER Diagram</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa antara kandidat <i>requirements</i> dan implementasi biaya | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa teknis <i>requirements</i> berdasarkan kebutuhan <i>user</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Diagram analisa <i>Fault Tree</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Hasil <i>requirements</i> secara detil | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Pemodelan <i>requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Analisa spesifikasi proses & sistem | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Model matematika Z | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Pemodelan <i>requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Hasil <i>Formal Requirements Inspection</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Hasil <i>Requirements Testing</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Hasil <i>Requirements Checklist</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Hasil <i>utility Test</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Data utilitas untuk <i>testing</i> | 0 | 0 | 0 | 0 | 0 | 0 |

Tabel 4. Derajat Nilai Atribut Teknik terhadap Teknik Rekayasa Kebutuhan [4].

| No. | Atribut Teknik | Teknik <i>Requirements Engineering</i> |
|-----|----------------|--|
|-----|----------------|--|

| | | <i>Brain Storming</i> | <i>Designer As Apprentice</i> | <i>Document Mining</i> | <i>Ethnography</i> | <i>Focus Group</i> | <i>Interview</i> |
|----|--|-----------------------|-------------------------------|------------------------|--------------------|--------------------|------------------|
| 1 | <i>Ability to facilitate the communication</i> | 0.8 | 0.8 | 0 | 0.6 | 1 | 0.8 |
| 2 | <i>Ability to understand social issues</i> | 0.4 | 1 | 0.8 | 0.8 | 1 | 0.8 |
| 3 | <i>Ability to get domain knowledge</i> | 1 | 1 | 1 | 1 | 0.6 | 0.6 |
| 4 | <i>Ability to get implicit knowledge</i> | 0.2 | 1 | 0.2 | 1 | 0.4 | 0.2 |
| 5 | <i>Ability to identify stakeholders</i> | 1 | 0.2 | 0.2 | 0.6 | 1 | 1 |
| 6 | <i>Ability to identify non-functional requirements</i> | 1 | 1 | 0.8 | 0.4 | 1 | 1 |
| 7 | <i>Ability to identify various viewpoints</i> | 0.8 | 0.2 | 0.4 | 0.4 | 0.8 | 0.8 |
| 8 | <i>Ability to model and understand requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | <i>Understanding ability for the notations used in analysis</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | <i>Ability to analyze non-functional requirements</i> | 0.6 | 0 | 0 | 0 | 0 | 0 |
| 11 | <i>Ability to facilitate the negotiation with customer</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | <i>Ability to prioritize the requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | <i>Ability to identify the accessibility of the system</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | <i>Ability to model interface requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | <i>Ability to identify and support requirements reuse</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | <i>Ability to represent requirements (expressibility)</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | <i>Capability for requirements verification</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | <i>Completeness of the semantics of the notation</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | <i>Ability to write unambiguous and precise requirements by using the notation</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | <i>Ability to write complete requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | <i>Capability for requirements management</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | <i>Modularity</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | <i>Implementability (Executability)</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | <i>Ability to identify the unambiguous requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | <i>Ability to identify the interaction (ambiguous, inconsistency, conflict)</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | <i>Ability to identify the incomplete requirements</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | <i>Ability to support COTS-based RE process</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | <i>Maturity of the supporting tool</i> | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | <i>Learning curve (Introduction cost)</i> | 0.2 | 0.2 | 0.2 | 0.4 | 0.6 | 0.2 |
| 30 | <i>Application cost</i> | 0.6 | 0.6 | 0.4 | 0.6 | 0.6 | 0.4 |
| 31 | <i>Complexity of the techniques</i> | 0.2 | 0.2 | 0.2 | 0.4 | 0.6 | 0.2 |

| | |
|------------------|---|
| Deskripsi Proyek | <i>This project is to develop a “Port Scheduling System” (PSS). The objective of the system is to schedule a container terminal with a throughput of maximum 1 million TEU (twenty foot equivalent unit) each year. The terminal must also be able to hand the smaller cargos. The project requires a highly interactive interface</i> |
| Atribut Proyek | <i>Project Size: Medium (>=500 and <1000 requirements) Project Complexity: Medium Requirements Volatility: Low Project Category: Semi-Detached Category Degree of Safety Criticality: Medium Time Constraints: Low Cost Constraints: Low Deliverables: Data permasalahan sistem saat ini, Analisa koleksi data dan requirements untuk sistem yang baru, Data Use Case dan actor (Desain dan Implementasi), Analisa teknis requirements berdasarkan kebutuhan user, Hasil requirements secara detil, Hasil Formal Requirements Inpection</i> |

Gambar 2. Masukan Atribut Proyek.

Metode MRETS

Setelah terpilih atribut teknik, kemudian dilakukan tahapan proses seleksi kombinasi teknik rekayasa kebutuhan menggunakan metode *MRETS* yang terdiri dari enam langkah yaitu:

1. Memberikan nilai pada atribut proyek yang diberikan.
2. Pemilihan atribut teknik rekayasa kebutuhan berdasarkan atribut proyek yang telah dipilih pada tahap satu. Derajat nilai atribut teknik terhadap teknik rekayasa kebutuhan dapat dilihat pada Tabel 4.
3. Seleksi kombinasi teknik rekayasa kebutuhan menggunakan metode *Fuzzy Clustering* berdasarkan atribut teknik rekayasa kebutuhan yang telah terpilih. Metode *Fuzzy Clustering* yang digunakan adalah metode *Fuzzy C-Means (FCM)*. *Algoritma FCM* adalah sebagai berikut:
 - a Masukan data yang akan dicluster X , berupa matriks berukuran $n \times b$ (n = jumlah teknik rekayasa kebutuhan, b = atribut teknik rekayasa kebutuhan).

X_{ij} = data teknik rekayasa kebutuhan ke- i ($i=1,2,\dots,n$), atribut teknik rekayasa kebutuhan ke- j ($j=1,2,\dots,b$).

- b. Dalam metode *FCM* ditentukan nilai awal lebih dulu sebagai berikut:

- Jumlah cluster = p
- Maksimum iterasi = $MaxIter$
- Error terkecil yang diharapkan= ϵ
- Objective Function = P
- Iterasi = t

- c Membuat bilangan random matriks u_{ik} dimana $i=1,2,\dots,n$ dan $k = 1,2,\dots,p$; sebagai elemen-elemen matriks partisi awal u .

- d Menghitung pusat cluster ke- k : m_{kj} , dengan $k = 1,2,\dots,p$; dan $j = 1,2,\dots,b$, sesuai Persamaan (1).

$$m_{ij} = \frac{\sum_{i=1}^n u_{ik}^2 X_{ij}}{\sum_{i=1}^n u_{ik}^2} \tag{1}$$

- e Menghitung perubahan matriks partisi, sesuai Persamaan (2).

$$u_{ik} = \frac{\left[\sum_{j=1}^b (X_{ij} - m_{ij})^2 \right]^{-1}}{\sum_{k=1}^p \left[\sum_{j=1}^b (X_{ij} - m_{ij})^2 \right]^{-1}} \tag{2}$$

4. Melakukan kalkulasi perhitungan berdasarkan *objective function* pada setiap iterasi ke-, sesuai Persamaan (3).

$$P_t = \sum \sum \left(\left[\sum (X_{ij} - m_{kj})^2 \right] u_{ik}^2 \right) \tag{3}$$

Kemudian akan ditentukan *Cost Function* sesuai Persamaan (4).

$$Cost = \sum_{k=1}^p \sum_{i=1}^n u_{ik}^2 d_{ik}^2 \tag{4}$$

Setelah jumlah cluster ditentukan kemudian dipilih cluster dengan anggota teknik rekayasa kebutuhan paling banyak dimana teknik rekayasa kebutuhan tersebut ditetapkan sebagai rekomendasi keluaran dari proses seleksi kombinasi dari metode *MRETS*.

5. Keluaran dari proses seleksi kombinasi teknik rekayasa kebutuhan menggunakan metode *MRETS* tersebut dihasilkan dalam bentuk *form report*, yaitu hasil seleksi teknik rekayasa kebutuhan yang direkomendasikan.
6. Melakukan evaluasi dari teknik rekayasa kebutuhan yang direkomendasikan. Hasil akhir teknik rekayasa kebutuhan mengacu kepada hasil semua tahap di atas dan pengalaman dari *engineers*. Beberapa teknik yang dihasilkan dapat tidak dipakai

apabila tidak sesuai dengan karakter atau domain permasalahan pada proyek tersebut. Pada penelitian ini, langkah no. 6 tidak dilakukan tetapi langsung dilakukan penerapan *Single Solution* untuk menentukan hasil teknik rekayasa kebutuhan.

6. *Rules* untuk mencari kombinasi teknik rekayasa kebutuhan yang mempunyai proses rekayasa kebutuhan tertinggi.

Dari semua tahapan *Rules* tersebut di atas akan menghasilkan teknik rekayasa kebutuhan yang akan digunakan dalam proyek perangkat lunak.

PENERAPAN SINGLE SOLUTION

Penerapan *Single Solution* bertujuan untuk efisiensi dan penyeragaman penentuan teknik rekayasa kebutuhan dari hasil rekomendasi teknik rekayasa kebutuhan yang dilakukan dengan metode *MRETS*. Penyeragaman diterapkan dengan menggunakan *Rules* sebagai acuannya. Penerapan *Single Solution* juga mengacu pada model proses rekayasa kebutuhan yang dikembangkan oleh *Kotonya dan Sommerville* [5].

Berikut ini adalah tahapan *Rules* yang digunakan dalam penerapan *Single Solution*:

1. *Rules* untuk mencari teknik rekayasa kebutuhan yang terpenuhi dan *deliverables* yang sudah terpenuhi.
2. *Rules* untuk mencari *deliverables* yang belum terpenuhi.
3. *Rules* untuk mencari *deliverables* yang dipunyai oleh hanya satu teknik rekayasa kebutuhan.
4. *Rules* untuk mencari kemungkinan kombinasi teknik rekayasa kebutuhan.
5. *Rules* untuk menghapus kombinasi yang *deliverables*nya tidak terpenuhi.

HASIL DAN PEMBAHASAN

Uji coba menggunakan data masukan atribut proyek dari penelitian sebelumnya seperti terlihat pada Gambar 2 dan ditambah data atribut proyek *deliverables*. Pada uji coba langkah pertama adalah menggunakan data masukan tersebut dan diproses sesuai tabel data atribut proyek terhadap data teknik rekayasa kebutuhan (Tabel 1 dan Tabel 3) dan tabel data atribut teknik terhadap data teknik rekayasa kebutuhan (Tabel 4). Pada Tabel 1 ditentukan teknik rekayasa kebutuhan yang mempunyai derajat nilai 0.75 dan 1 terhadap atribut proyek berdasarkan masukan pada Gambar 2, demikian juga pada Tabel 3 ditentukan teknik rekayasa kebutuhan yang mempunyai derajat nilai 0.75 dan 1 terhadap *deliverables* berdasarkan masukan pada Gambar 2.

Setelah terpilih teknik rekayasa kebutuhan, maka dilanjutkan uji coba langkah kedua yaitu menggunakan data pada Tabel 4 untuk memilih atribut teknik. Pada Tabel 4 ditentukan atribut teknik yang mempunyai derajat nilai 0.6, 0.8, dan 1 terhadap teknik rekayasa kebutuhan terpilih. Proses tersebut menghasilkan data atribut teknik rekayasa kebutuhan sebanyak dua puluh sembilan atribut teknik.

Tabel 5. Teknik Rekayasa Kebutuhan Berdasarkan Pemetaan Masukan *Deliverables* dan Model *Kotonya dan Sommerville*.

| <i>Elicitation</i> | Proses <i>Requirements Engineering</i> | | |
|--|--|--|--|
| | <i>Analysis & Negotiation</i> | <i>Documentation</i> | <i>Verification & Validation</i> |
| <i>Evolutionary Prototypes Scenario Approach</i> | <i>Scenario Approach</i> | <i>Structured Natural Language specification Problem Frame Oriented Analysis</i> | <i>Scenario Approach</i> |
| <i>Representation Modeling Fault Tree Analysis</i> | <i>Goal-Oriented Analysis Object-Oriented Analysis Representation Modeling Functional Decomposition Problem Frame Oriented Analysis Fault Tree Analysis XP</i> | | <i>Formal Requirements Specification Problem Frame Oriented Analysis</i> |

Tabel 6. Kombinasi Teknik Rekayasa Kebutuhan Terpilih Penerapan *Single Solution*.

| Proses Requirements Engineering | | | |
|---------------------------------|--|--|--|
| <i>Elicitation</i> | <i>Analysis & Negotiation</i> | <i>Documentation</i> | <i>Verification & Validation</i> |
| <i>Evolutionary Prototypes</i> | <i>Object-Oriented Analysis</i> <i>XP</i> | <i>Structured Natural Language spesification</i> | <i>Formal Requirements Spesification</i> |

Pada uji coba langkah ketiga, hasil atribut teknik tersebut diproses dengan menggunakan metode *MRETS* dan menghasilkan teknik rekayasa kebutuhan terpilih. Hasil teknik terpilih tersebut dipetakan dengan *deliverables input* (*deliverables* mempunyai nilai satu) sesuai Tabel 3 dan menghasilkan sepuluh teknik rekayasa kebutuhan yaitu: *Evolutionary Prototypes, Scenario Approach, Goal-Oriented Analysis, Representation Modeling, Functional Decomposition, Problem Frame Oriented Analysis, Fault Tree Analysis, Object-Oriented Analysis, Structured Natural Language Spesification*, dan *XP*. Kemudian teknik tersebut dipetakan sesuai model proses rekayasa kebutuhan yang dikembangkan oleh *Kotonya dan Sommerville* [5] yang berisi empat tahapan yaitu: *requirements elicitation, requirements analysis and negotiation, requirements documentation*, dan *requirements verification and validation* sesuai Tabel 5.

Pada uji coba langkah keempat, hasil teknik rekayasa kebutuhan di atas tersebut diformulasikan dalam penerapan *Single Solution* sehingga menghasilkan empat puluh tujuh kombinasi. Dari seluruh kombinasi tersebut yang memenuhi seluruh *Rules* dalam penerapan *Single Solution* adalah teknik:

Evolutionary Prorotypes, Object-Oriented Analysis, XP, Structured Natural Language Spesification, dan *Formal Requirements Spesification*. Kombinasi terpilih tersebut dapat dilihat pada Tabel 6.

Dari uji coba di atas dapat dianalisa bahwa penerapan *Single Solution* mengurangi penggunaan teknik rekayasa kebutuhan. Penerapan *Single Solution* juga berguna sebagai penyeragaman hasil proses seleksi kombinasi teknik rekayasa kebutuhan.

SIMPULAN

Dari hasil penelitian penambahan masukan atribut proyek *Deliverables* dan penerapan *Single Solution* berbasis metode *MRETS* dapat diambil simpulan sebagai berikut:

1. Penelitian ini memberikan kontribusi terhadap efisiensi penggunaan teknik rekayasa kebutuhan. Hal ini disebabkan *Rules* yang digunakan dapat meminimalisasi pemilihan teknik rekayasa kebutuhan.
2. Keluaran dari penelitian ini merupakan hasil obyektif dari sistem. Dalam uji coba yang telah dilakukan, keluaran teknik rekayasa kebutuhan yang dihasilkan mempunyai hasil yang tetap bila diberikan masukan atribut proyek yang sama.

DAFTAR PUSTAKA

- [1] The Standish Group International Inc, The Chaos Report. 1995. URL: http://www.standishgroup.com/sample_research/index.php. diakses tanggal 5 Januari 2009.
- [2] Alan DM. *Software Requirements, Objects: Functions and States*, New Jersey: Prentice Hall. 1993.
- [3] Glass RL and Vessey I. Contemporary Application-Domain Taxonomies. *IEEE Software*. 63-76. 1995.
- [4] Li J, Eberlein A and Far BH. Combining Requirements Engineering Techniques-Theory and Case Study. *Proceedings IEEE ECBS*. 05:12. 2005.
- [5] Kotonya G and Sommerville I. *Requirements Engineering: Processes and Techniques 3rd Edition*. New York: John Wiley and Sons Ltd. 1998.