

Komparasi Kualitas Standar Dokumen SRS: MIL-STD-498 dan IEEE 830-1998

Bimo Prakoso¹, Hanson Prihantoro Putro²

Program Studi Teknik Informatika
FTI, Universitas Islam Indonesia
Yogyakarta

¹13523078@students.uui.ac.id, ²hanson@uui.ac.id

Abstrak—Hasil proses analisis kebutuhan perangkat lunak dituangkan ke dalam dokumen *Software Requirements Specification* (SRS). Untuk membuat dokumen tersebut, terdapat beragam standar penulisan yang bisa digunakan. Banyaknya standar yang tersedia seringkali membingungkan penggunaannya ketika harus memilih standar yang tepat untuk mereka. Karena itu perlu dilakukan komparasi berbagai standar penulisan yang ada untuk mengetahui standar yang bisa menyediakan dokumen SRS dengan kualitas lengkap, ringkas, dan tidak sulit untuk diaplikasikan oleh pengembang pemula di Indonesia. Standar yang dipilih untuk dikomparasi adalah IEEE 830-1998 dan MIL-STD-498 (DI-IPSC-81433). Komparasi dilakukan dengan mencari perbedaan antara kedua dokumen panduan dan dokumen SRS yang dihasilkan, menggali kelebihan dan kekurangannya, serta mengujikan dokumen SRS tersebut kepada pihak ahli. Dari proses komparasi yang dilakukan, diperoleh standar pembuatan dokumen SRS yang memenuhi kualitas ringkas, lengkap, dan mudah diterapkan, yaitu standar IEEE 830-1998.

Kata kunci—komparasi; SRS; IEEE 830-1998; MIL-STD-498.

I. PENDAHULUAN

Pada fase pengembangan perangkat lunak, pihak analis terlebih dahulu akan mengumpulkan berbagai kebutuhan perangkat lunak yang diinginkan oleh pelanggan. Proses ini penting dan harus dilakukan secara benar sebab kesalahan pada fase ini akan menyebabkan biaya perbaikan yang sangat besar dan proses pengembangan perangkat lunak pun menjadi lebih lama.

Hasil analisis kebutuhan yang telah diperoleh, selanjutnya akan dimuat pada dokumen *Software Requirements Specification* (SRS). Dokumen ini menjelaskan secara lengkap mengenai maksud dan tujuan dari sistem yang akan dibangun beserta lingkungan di mana sistem itu hendak dioperasikan [1]. Dalam membuat dokumen SRS terdapat berbagai standar yang bisa digunakan sebagai acuan, seperti: 1) Standar keluaran IEEE seri 830 tahun 1993 atau 1998; 2) ISO/IEC/IEEE 29148:2011 yang merupakan standar pengganti dari IEEE 830-1998; atau 3) MIL-STD-498 (DI-IPSC-81433) yang dikeluarkan oleh Departemen Pertahanan Amerika Serikat.

Banyaknya standar yang tersedia seharusnya membuat proses pembuatan dokumen SRS menjadi lebih mudah. Akan tetapi karena banyaknya pilihan akan standar yang ada, hal ini seringkali membuat penggunaannya kesulitan dalam menentukan standar yang tepat.

Oleh karena itu, perlu dilakukan komparasi terhadap standar-standar yang ada untuk mengetahui standar yang dapat menghasilkan dokumen SRS dengan kualitas lengkap, ringkas, dan mudah diterapkan oleh pengembang pemula. Standar yang dipilih untuk dikomparasi adalah IEEE 830-1998 dan MIL-STD-498 (DI-IPSC-81433). Dipilihnya kedua standar ini dengan alasan, walau ISO/IEC/IEEE 29148:2011 merupakan pengganti dari IEEE 830-1998, namun standar tersebut masih jarang diaplikasikan. Berbeda dengan IEEE 830-1998 yang masih banyak digunakan sehingga mudah ditemukan referensinya. Selain itu standar ini juga digunakan oleh perguruan tinggi tempat penelitian berlangsung sebagai rujukan dalam membuat dokumen SRS pada matakuliah Rekayasa Perangkat Lunak. Maka dari itu, standar ini kemudian dipilih untuk dikomparasi. Sedangkan MIL-STD-498 sendiri dipilih karena standar ini biasa digunakan untuk membangun perangkat lunak yang memiliki tingkat kerumitan dan bahaya yang tinggi. Umumnya standar ini digunakan oleh pihak militer atau pengembang yang membuat perangkat lunak untuk kebutuhan militer.

Penelitian ini bertujuan untuk menghasilkan dokumen SRS dari panduan yang dikeluarkan oleh kedua standar. Keduanya akan digunakan untuk membuat dokumen SRS untuk pengembangan perangkat lunak *Job Order Request* di sebuah perusahaan industri di Kalimantan Timur. Kemudian penelitian dilanjutkan dengan membandingkan hasil dari kedua SRS tersebut dengan membandingkan juga panduan asli dari keduanya. Diharapkan dari penelitian ini, para pengembang perangkat lunak mendapatkan panduan yang lebih mudah dan nyaman dalam membuat sebuah dokumen SRS.

II. PEKERJAAN TERKAIT

Requirement engineering merupakan proses menghimpun, mencerna, dan memastikan setiap kebutuhan yang diperlukan oleh pengguna atau pelanggan. Hasil dari proses ini selanjutnya akan dimuat pada dokumen SRS [2]. Dokumen SRS dapat dihasilkan oleh seluruh pihak yang terlibat dalam proses pengembangan perangkat lunak seperti pelanggan, pengguna, maupun pengembang [3]. Dokumen ini menjadi alat komunikasi antara pihak pelanggan dan pengembang agar masing-masing pihak dapat mengetahui secara pasti kebutuhan dari perangkat lunak yang akan dikembangkan dan bisa memvalidasi kebenarannya.

Terdapat berbagai standar yang bisa dirujuk untuk membantu proses pembuatan dokumen ini, seperti IEEE 830-1998 [4] dan MIL-STD-498 (DI-IPSC-81433) [5]. Kedua standar tersebut akan dikomparasi untuk mengetahui standar yang dapat menyajikan dokumen SRS dengan kualitas ringkas, lengkap, dan mudah diterapkan oleh pengembang pemula. Selain ketiga kriteria kualitas tersebut, kriteria lain juga akan digunakan seperti kelengkapan, keringkasannya, kebenaran, ambiguitas beserta beberapa kriteria lainnya [3]. Terdapat beberapa pekerjaan yang mengkaji penggunaan kedua standar ini, baik untuk IEEE 830-1998 [6][7][8] maupun untuk MIL-STD-498 [9][10]. Namun demikian, artikel-artikel tersebut hanya memuat terkait penggunaan standar tersebut.

Terdapat sebuah artikel yang melakukan komparasi terhadap empat standar berbeda yaitu IEEE/EIA 12207, ISO/IEC 12207, J-STD-016, dan MIL-STD-498. Pada artikel tersebut, komparasi yang dilakukan lebih ditujukan untuk mencari perbedaan dan persamaan dari masing-masing standar beserta proses awal mula standar-standar tersebut dihasilkan. Akan tetapi proses komparasinya tidak sampai pada tahap mengimplementasikan ke empat standar tersebut. Kajian yang dilakukan juga belum begitu mendalam [11]. Belum ada penelitian yang melakukan perbandingan terhadap standar-standar terkait dokumen pengembangan perangkat lunak, terlebih dokumen SRS.

Sebagai studi kasus, masing-masing dokumen SRS akan membahas studi kasus pengembangan perangkat lunak *Job Order Request* (JOR) di PT. Pupuk Kaltim. JOR merupakan aplikasi untuk melakukan permintaan pelayanan dengan mengirimkan pekerja untuk menyelesaikan suatu permasalahan [12]. Dengan adanya JOR, setiap karyawan perusahaan dapat mengajukan permintaan layanan yang mereka butuhkan kepada perusahaan melalui Departemen Pelayanan Umum (DPU). Nantinya, seluruh permintaan tersebut akan ditampung oleh DPU dan dikirimkan ke instansi-intansi yang bertugas untuk memenuhi atau menyelesaikan permintaan tersebut. Beberapa layanan yang disediakan oleh JOR seperti permohonan pinjaman perlengkapan acara, sarana pertemuan, perbaikan listrik, AC, *plumbing*, dan lain sebagainya.

III. METODOLOGI

Terdapat empat tahapan yang harus dilalui ketika melakukan komparasi pada penelitian ini (Gambar 1). Tahap pertama yaitu menentukan *template* dokumen yang akan digunakan. Tahap kedua adalah fase pembuatan dokumen SRS. Tahap ketiga pengujian dokumen SRS. Kemudian tahap terakhir yaitu mengevaluasi hasil komparasi.



Gambar 1 Tahap komparasi

Pada penelitian ini *template* yang digunakan untuk membuat dokumen SRS akan mengacu pada standar IEEE (Gambar 2) dan MIL-STD-498 (Gambar 3). *Template* tersebut perlu dilakukan perubahan di beberapa bagian agar dapat sesuai dengan studi kasus yang diangkat. Khusus IEEE, karena standar ini memiliki 8 *template* tambahan untuk mengatur bab *specific requirements*,

template fitur dipilih lantaran kesesuaiannya dengan perangkat lunak yang dikembangkan.

1. Introduction
1.1 Purpose
1.2 Scope
1.3 Definitions, acronyms, and abbreviations
1.4 References
1.5 Overview
2. Overall description
2.1 Product perspective
2.2 Product functions
2.3 User characteristics
2.4 Constraints
2.5 Assumptions and dependencies
3. Specific requirements
3.1 External interface
3.2 Function
3.3 Performance requirements
3.4 Logical database requirement
3.5 Design constraints
3.5.1 Standards compliance
3.6 Software system attributes
3.6.1 Reliability
3.6.2 Availability
3.6.3 Security
3.6.4 Maintainability
3.6.5 Portability
3.7 Additional comments
Appendix
Index

Gambar 2 *Template* dokumen SRS IEEE 830-1998 [4]

1. Scope
1.1 Identification
1.2 System overview
1.3 Document overview
2. Referenced documents
3. Requirements
3.1 Required states and modes
3.2 CSCI capability requirements
3.2.x (CSCI capability)
3.3 CSCI external interface requirements
3.3.1 Interface identification and diagrams
3.3.x (Project-unique identifier of interface)
3.4 CSCI internal interface requirements
3.5 CSCI internal data requirements
3.6 Adaptation requirements
3.7 Safety requirements
3.8 Security and privacy requirements
3.9 CSCI environment requirements
3.10 Computer resource requirements
3.10.1 Computer hardware requirements
3.10.2 Computer hardware resource utilization requirements
3.10.3 Computer software requirements
3.10.4 Computer communications requirements
3.11 Software quality factors
3.12 Design and implementation constraints
3.13 Personnel-related requirements
3.14 Training-related requirements
3.15 Logistics-related requirements
3.16 Other requirements
3.17 Packaging requirements
3.18 Precedence and criticality of requirements
4. Qualification provisions
5. Requirements traceability
6. Notes
A. Appendixes

Gambar 3 *Template* dokumen SRS MIL-STD-498 [5]

Selanjutnya apabila proses pembuatan dokumen SRS telah selesai, kedua dokumen tersebut akan diuji terkait hal:

- Standar yang biasa digunakan oleh pihak ahli
- Kualitas dari dokumen SRS yang dihasilkan, dan
- Kesalahan pada dokumen SRS

Pengujian dilakukan kepada dua staf pengajar Rekayasa Perangkat Lunak dengan menggunakan kuesioner dan parameter yang diujikan untuk kedua dokumen tersebut adalah [3]:

- Standar yang umum digunakan
Parameter ini digunakan untuk mencari tahu standar yang biasa dipakai oleh pihak ahli ketika membuat dokumen SRS.
- Kebenaran
Parameter ini digunakan untuk memeriksa kebenaran dan kesesuaian dokumen SRS yang telah dihasilkan, dibandingkan dengan panduan dari standar yang digunakan.
- Kecukupan
Parameter ini digunakan untuk memastikan dokumen SRS yang dihasilkan sudah cukup memenuhi syarat penulisan suatu dokumen SRS.
- Ambiguitas
Parameter ini digunakan untuk memastikan setiap kebutuhan di masing-masing dokumen SRS tidak ada yang ambigu, mudah dipahami, dan jelas.
- Bisa dimengerti
Parameter ini digunakan untuk mencari tahu standar yang lebih mudah dimengerti oleh penguji, baik dari pedomannya maupun hasilnya.
- Keringkasan
Parameter ini digunakan untuk memastikan setiap kebutuhan pada dokumen SRS sudah dinyatakan secara jelas dan singkat.
- Kelengkapan
Parameter ini digunakan untuk memastikan komponen-komponen minimal dari sebuah SRS sudah dimuat di masing-masing dokumen.
- Keringkasan dan kelengkapan
Parameter ini digunakan untuk mencari tahu standar yang memenuhi kualitas ringkas namun tetap lengkap secara sekaligus.
- Kesalahan dan kekurangan pada dokumen
Parameter ini digunakan untuk menemukan beragam kesalahan pada dokumen SRS yang ditemui oleh pihak penguji.

IV. HASIL DAN PEMBAHASAN

A. Hasil Dokumen SRS

Dari proses pembuatan dokumen SRS yang telah dilakukan, ditemukan sejumlah perbedaan baik pada dokumen panduan maupun dokumen SRS di masing-masing standar. Untuk mengetahui apa saja perbedaan tersebut dapat dilihat pada Tabel 1.

TABEL 1 HASIL KOMPARASI BERDASARKAN PANDUAN DAN DOKUMEN SRS

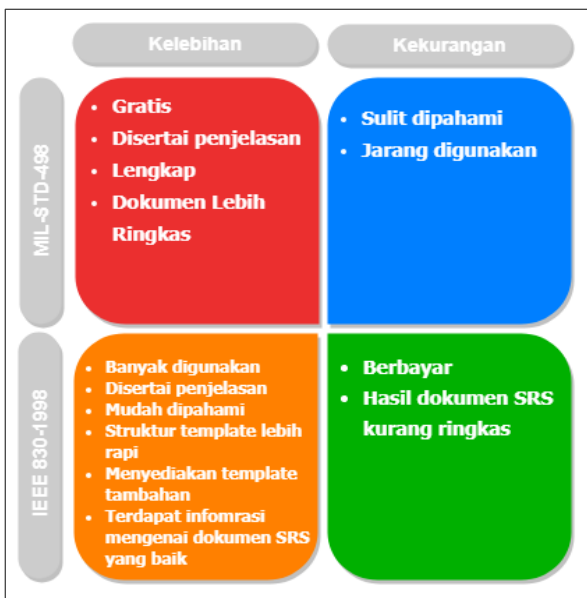
No.	MIL-STD-498	IEEE 830-1998
1.	Dokumen terdiri atas 6 bab.	Dokumen terdiri atas 4 bab.
2.	Jumlah halaman sebanyak 24 halaman.	Jumlah halaman sebanyak 47 halaman.
3.	Fungsi-fungsi perangkat lunak dimuat pada subbab "Kebutuhan Kemampuan Perangkat Lunak"	Fungsi-fungsi perangkat lunak dimuat pada subbab "Fungsionalitas"
4.	Berbagai fungsi perangkat lunak dikelompokkan menjadi modul-modul.	Berbagai fungsi perangkat lunak disusun berdasarkan fungsionalitasnya.
5.	Nomor versi perangkat lunak harus dimuat.	Nomor versi perangkat lunak tidak harus dimuat.
6.	Skenario masukan/keluaran dari masing-masing fungsi tidak harus dimuat.	Skenario masukan/keluaran dari masing-masing fungsi harus dimuat.
7.	Kebutuhan <i>state</i> dan mode tetap dimuat pada dokumen walupun sistem tidak memiliki kebutuhan tersebut.	Kebutuhan <i>state</i> tidak dikenal. Selain itu pengguna juga tidak wajib untuk memuat kebutuhan mode karena kebutuhan tersebut diatur pada <i>template</i> yang berbeda.
8.	Subbab 1.2 mengidentifikasi pihak-pihak yang akan menjadi target pengguna dari sistem. Sedangkan subbab 1.3 memuat informasi mengenai karakteristik dari pengguna sistem.	Karakteristik umum dari pengguna sistem wajib dijelaskan. Informasi mengenai target pengguna sistem tetap dimuat walaupun tidak terdapat aturan yang mewajibkan hal tersebut.
9.	Informasi mengenai target pembaca dokumen tetap ditambahkan meskipun tidak terdapat aturan yang mengharuskan hal tersebut.	Informasi mengenai siapa saja target pembaca dokumen diatur pada standar ini dan dimuat pada subbab 1.1.
10.	Daftar akronim dan glosarium dimuat pada akhir dokumen.	Daftar definisi dan singkatan dimuat pada awal dokumen.
11.	Tidak terdapat informasi mengenai kebutuhan performa.	Terdapat informasi mengenai kebutuhan performa.
12.	Bab 1 hanya terdiri dari 3 subjek bahasan dan bab 2 hanya 1 subjek bahasan.	Bab 1 terdiri dari 5 subjek bahasan dan bab 2 juga 5 bahasan.
13.	Tidak tersedia variasi <i>template</i> untuk menyusun bab 3.	Tersedia 8 variasi <i>template</i> yang dapat dipilih guna menyusun bab 3.
14.	Terdapat kebutuhan antarmuka eksternal dan internal.	Hanya terdapat kebutuhan antarmuka eksternal.
15.	Menggunakan istilah "Kebutuhan" untuk menyatakan subbab kebutuhan perangkat keras komputer, kebutuhan perangkat lunak komputer, serta kebutuhan komunikasi komputer. Informasi tersebut dimuat pada subbab "Kebutuhan Sumber Daya Komputer".	Menggunakan istilah "Antarmuka" untuk menyatakan subbab antarmuka perangkat keras, antarmuka perangkat lunak, serta antarmuka komunikasi. Informasi tersebut dimuat pada subbab Kebutuhan Antarmuka Eksternal.
16.	Terdapat kebutuhan keamanan dan kerahasiaan seperti pada subbab 1.3 "Ulasan Dokumen" serta subbab 3.8 "Persyaratan keamanan dan kerahasiaan"	Hanya terdapat kebutuhan keamanan saja.

Secara umum, perbedaan yang paling terlihat antara standar MIL dan IEEE ada di jumlah halamannya. Dengan studi kasus yang sama, dokumen SRS yang dihasilkan oleh standar MIL jauh lebih ringkas daripada dokumen SRS yang dihasilkan oleh IEEE walaupun MIL memiliki jumlah bab yang lebih banyak.

Perbedaan-perbedaan yang ditemui di kedua dokumen ini sebisa mungkin dibuat seragam agar memudahkan proses pembuatan dokumen dan proses komparasinya. Namun beberapa perbedaan dipilih untuk tidak diseragamkan karena struktur maupun aturan di kedua standar tersebut memang tidak sama. Oleh sebab itu, masih dapat dilihat sejumlah perbedaan di kedua dokumen.

B. Kelebihan dan Kekurangan

Baik MIL-STD-498 maupun IEEE 830-1998 memiliki sejumlah kelebihan dan kekurangan. Gambar 4 menampilkan daftar kelebihan dan kekurangan tersebut.



Gambar 4 Kelebihan dan kekurangan MIL-STD-498 dan IEEE 830-1998

MIL-STD-498 dapat diperoleh dan digunakan siapa saja secara gratis karena sifatnya yang *public domain*. Selain itu MIL juga sangat lengkap sebab standar ini telah menyediakan seluruh *template* dokumentasi yang diperlukan selama aktivitas pengembangan perangkat lunak berlangsung. *Template* yang diberikan juga tidak hanya berupa struktur dokumennya saja melainkan juga disertai dengan penjelasan di setiap subjek bahasannya. Selain itu dokumen yang dihasilkan oleh MIL-STD-498 juga lebih ringkas, karena tidak terdapat skenario masukan/keluaran di masing-masing kebutuhan fungsionalitas dan subjek pembahasan yang dimuat pada bab 1 dan bab 2-nya juga tidak sebanyak IEEE. Akan tetapi, penjelasan yang diberikan standar ini tidak mudah untuk dipahami dan cenderung membingungkan. Kurangnya penjelasan eksplisit merupakan kelemahan yang paling tampak dari standar MIL-STD-498 ini. Selain itu referensi yang ditemukan juga tidak banyak membantu proses pembuatan dokumen SRS sebab pembahasan yang terdapat pada referensi tersebut cukup sulit untuk dimengerti dan kurang familiar bagi pengembang pemula mengingat rujukan tersebut lebih banyak membahas perangkat

lunak di bidang militer atau sistem dengan tingkat kerumitan dan bahaya yang tinggi.

Sedangkan kelebihan yang dimiliki IEEE adalah terkait panduannya banyak digunakan sehingga referensi mengenai *best practice* ini sangat mudah ditemukan. Kemudian sama seperti MIL-STD-498, *template* yang diberikan juga tidak hanya berupa struktur dokumennya saja namun juga disertai dengan penjelasan di setiap subjek bahasannya. Selain itu penjelasan yang diberikan oleh IEEE juga lebih mudah dipahami ketimbang MIL-STD-498 dan tersedia berbagai informasi yang harus menjadi perhatian pengembang dalam membuat dokumen SRS. Kemudian tersedia juga 8 alternatif *template* tambahan yang bisa dipilih sesuai dengan kebutuhan. Hal ini agar informasi yang dimuat pada dokumen SRS dapat sesuai dengan karakter perangkat lunak yang dibangun. Dokumen SRS-nya pun bisa menjadi lebih rinci, lengkap dan mudah dimengerti oleh pembaca. Struktur *template* dokumen yang disediakan oleh IEEE pun cukup rapi dan detail sebab informasi terkait dokumen dan sistem ada di bagian yang berbeda sehingga proses pencarian informasi dari sisi pembaca pun menjadi lebih mudah. Namun, karena standar ini tidak bersifat *public domain* maka untuk memperolehnya kita harus membelinya terlebih dulu. Selain itu kekurangan lain dari standar ini yaitu hasil dokumen SRS-nya tidak seringkas dokumen SRS yang menggunakan standar MIL-STD-498.

C. Pengujian Dokumen SRS

Terdapat dua pihak yang terlibat dalam pengujian dokumen ini, yaitu staf pengajar sebagai ahli dan mahasiswa. Dari hasil pengujian yang dilakukan oleh pihak ahli, sejumlah kekurangan masih ditemukan baik pada dokumen SRS yang menggunakan format MIL maupun IEEE. Namun, secara umum kedua dokumen tersebut sudah cukup memenuhi standar penulisan dokumen SRS menurut mereka. Selain itu pihak ahli juga mengungkapkan bahwa dokumen SRS yang menggunakan format IEEE hasilnya lebih lengkap dan lebih mudah untuk dipahami.

Untuk mengetahui berbagai kekurangan yang ditemui pihak ahli pada kedua dokumen SRS, dapat dilihat pada Gambar 5. Dari gambar tersebut diketahui bahwa standar MIL-STD-498 menghasilkan lebih banyak kesalahan daripada standar IEEE 830-1998. Berbagai kekurangan maupun kesalahan tersebut perlu menjadi perhatian pengembang pemula sewaktu mereka hendak memilih standar MIL-STD-498 maupun IEEE 830-1998 sebab kekurangan-kekurangan yang ada pada Gambar 5 sangat mungkin terjadi ketika pengembang pemula menggunakan salah satu dari kedua standar tersebut. Detil hasil kuesioner pengujian dapat dilihat pada lampiran.

MIL-STD-498	IEEE 830-1998	MIL-STD-498 & IEEE 830-1998
Detail kebutuhan kualitas perangkat lunak masih kurang tepat	Informasi terkait kebutuhan fungsionalitas muncul bekal-kali.	Karakteristik antarmuka eksternal tidak lengkap dan bentuk antarmuka eksternal akan seperti apa?
Kebutuhan keamanan seharusnya diletakkan pada subbab persyaratan keamanan dan privasi	Tidak ada penjelasan cara koneksi perangkat lunak ke basisdata pusat (API atau LDAP)	Tidak ada penjelasan terkait penanganan masalah dengan entitas luar, seperti <i>basisdata down</i> , dsb
Tidak ada penjelasan terkait standar keamanan IT di PT. Pupuk Kaltim	Tidak <i>Leveling</i> kebutuhan (kebutuhan prioritas)	Detil standar pengembangan di organisasi kurang digali
Tidak ada spesifikasi kebutuhan terkait perangkat keras komputer	Tidak terdapat rincian subbab perspektif produk seperti pada panduan	Tidak tersedia notasi diagram seperti <i>usecase</i> diagram, UML, ERD, dsb
Tidak terdapat alasan antarmuka internal perangkat lunak diserahkan ke programmer	Tidak terdapat rincian subbab batasan seperti pada panduan	Tidak terdapat penomoran unik di sejumlah kebutuhan
Tidak ada subbab 3.3.x (a, b, c, d, e) seperti di panduan	<p>Keterangan:</p> <p>■ Daftar kekurangan yang telah diperbaiki</p> <p>■ Daftar kekurangan yang belum dapat diperbaiki</p>	

Gambar 5 Daftar kekurangan dokumen SRS beserta umpan baliknya

Hasil yang sama terkait preferensi juga diperoleh ketika pengujian dilakukan kepada sejumlah mahasiswa. Mereka mengatakan bahwa dokumen SRS yang menggunakan format IEEE ternyata lebih lengkap. Hal itu dikarenakan informasi terkait prototipe antarmuka pengguna dan skenario dari masing-masing fungsionalitas telah tersedia, sehingga memberikan mereka gambaran yang lebih jelas tentang sistem yang akan dikembangkan. Selain itu menurut pihak mahasiswa dokumen SRS versi IEEE dan dokumen panduannya juga lebih nyaman untuk dibaca dan mudah dipahami walaupun jumlah halaman yang dihasilkan oleh standar ini jauh lebih banyak sehingga dibutuhkan waktu yang tidak sedikit untuk membacanya.

Berbeda dengan IEEE, dokumen SRS versi MIL-STD-498 justru menghasilkan jumlah halaman yang jauh lebih sedikit ketimbang IEEE. Hal ini menjadi nilai positif dari standar ini menurut pihak mahasiswa. Akan tetapi dokumen panduannya cenderung sulit untuk dimengerti dan dokumen SRS yang dihasilkan juga tidak lengkap, sehingga pihak mahasiswa kurang memahami sistem seperti apa yang ingin dikembangkan.

V. KESIMPULAN DAN SARAN

Komparasi telah berhasil dilakukan terhadap dokumen panduan dan dokumen SRS baik dari versi MIL-STD-498 maupun IEEE 830-1998. Hasil komparasi tersebut menunjukkan

bahwa sejumlah perbedaan masih ditemukan di masing-masing standar, seperti perbedaan dari sisi struktur dokumen, subjek bahasan, ataupun beberapa istilah teknis yang digunakan. Perbedaan-perbedaan itu pada akhirnya mempengaruhi faktor keringkas dan kelengkapan dokumen SRS yang dihasilkan terutama perbedaan pada struktur dokumen dan subjek bahasanya.

Bagi pengembang pemula standar IEEE 830-1998 sangat disarankan untuk digunakan sebab dokumen SRS yang dihasilkan dari standar ini memiliki informasi yang jauh lebih lengkap, mudah dimengerti, dan nyaman untuk dibaca meskipun hasil dokumennya tidak seringkis dokumen SRS versi MIL-STD-498. Selain itu panduan yang disediakan oleh IEEE juga lebih mudah untuk dipahami, dan referensinya tidak sulit untuk dicari.

Untuk penelitian berikutnya saran yang dapat diberikan adalah, mengkaji kembali beberapa kekurangan dokumen SRS yang belum bisa diperbaiki oleh penulis (lihat Gambar 5). Kemudian melakukan komparasi terhadap standar pembuatan dokumen SRS lainnya seperti ISO/IEC/IEEE 29148:2011 atau DI-IPSC-81433A (MIL-STD-498) untuk mengetahui kelebihan dan kekurangan dari standar penulisan dokumen SRS yang berbeda. Menggunakan teknik pengujian *usability* juga sangat disarankan agar kita dapat memiliki gambaran mengenai seperti apa dokumen SRS yang telah dihasilkan. Namun, untuk menggunakan metode tersebut perlu dirumuskan kembali skenario yang lebih efektif guna menghindari rasa jenuh dan kelelahan dari pihak partisipan selama proses pengujian berlangsung. Kemudian perlu dibuat *template* dokumen SRS versi baru dengan menggabungkan *template* dari MIL-STD-498 dan IEEE 830-1998 agar dokumen SRS yang dihasilkan dapat memenuhi kualitas yang lebih ringkas, lengkap dan mudah.

REFERENSI

- [1] M. Rouse, "Definition Software Development," *WhatIs.com*, Oktober 2016. [Online]. Available: <http://whatis.techtarget.com/definition/software-development>. [Accessed May 3, 2017]
- [2] R. S. Wahono, "Menyegarkan Kembali Pemahaman Tentang Requirements Engineering," *RomiSatriaWahono.net*, April 29 2006. [Online]. Available: <http://romisatriawahono.net/2006/04/29/menyegarkan-kembali-pemahaman-tentang-requirement-engineering>. [Accessed May 8, 2017].
- [3] M. J. Ali, "Metrics for Requirements," Master Thesis, Umea University Swedia, 2006.
- [4] *IEEE Recommended Practice for Software Requirements Specifications, IEEE Xplore*, IEEE Computer Society, Software Engineering Standards Committee, and IEEE-SA Standards Board, 1998.
- [5] *Software Requirement Specification (DI-IPSC-81433)*, Department of Defense United States of America, 1994.
- [6] M. Suzanna and A. H. N. Ali, "Analisis Kebutuhan untuk Pengembangan Sistem Informasi Produksi di PT Z dengan Metode VORD (Viewpoints Oriented Requirements Definition)," in *Prosiding Seminar Nasional Manajemen Teknologi XIV*, Surabaya, 2011.
- [7] Y. Kurniawan and Yuswanto, "Software Requirement Specification Sistem Perencanaan Biaya Perjalanan Ibadah Haji Sesuai dengan Standard IEEE 830-1998," *Prosiding Konferensi Nasional Sistem Informasi*, 2014.
- [8] E. Yağdereli, C. Gemci and A. Z. Aktaş, "A study on cyber-security of autonomous and unmanned vehicles," *The Journal of Defense Modeling and Simulation*, vol. 12, p. 4, pp. 369-381, 2015.

[9] J. R. Schwartz, S. Mark and A. Wolfson, "A first-order simulator to control dioxin emissions: NMCRC-ATMOS," *Waste Management & Research*, vol. 28, no. 5, pp. 461-471, 2010.

[10] N. Abbas, "Software quality and governance in agile software development," Doctoral Dissertation, University of Southampton UK, 2009.

[11] L. Gray, "A Comparison of IEEE/EIA 12207, ISO/IEC 12207, J-STD-016, and MIL-STD-498 for Acquirers and Developers," *Abelia Corporation*, Fairfax, Virginia, USA, 1999.

[12] O. I. Nugraha and K. Afandi, "Rancang Bangun Sistem Informasi Job Order Request Berbasis Web Dengan Menggunakan Aplikasi Joget Workflow," Laporan Kerja, Bontang, 2016.

LAMPIRAN: DATA HASIL KUESIONER PENGUJIAN DOKUMEN SRS.

No.	Pertanyaan	Penguji 1		Penguji 2	
		MIL-STD-498	IEEE 830-1998	MIL-STD-498	IEEE 830-1998
1.	Dalam membuat dokumen SRS manakah panduan yang paling sering anda gunakan atau familier menurut anda?	✓	✓		✓
2.	Apakah anda tahu atau pernah mendengar panduan pembuatan dokumen SRS MIL-STD-498 dan IEEE 830-1998 sebelum terlibat dalam pengujian ini?	Ya	Ya	Tidak	Ya
3.	Apakah anda pernah menggunakan kedua panduan tersebut ketika membuat dokumen SRS baik dalam proses pengembangan suatu perangkat lunak maupun proses belajar mengajar di kampus?	Ya	Ya	Tidak	Ya
4.	Semua informasi/kebutuhan di dokumen SRS sudah dimuat pada subbab yang tepat.	Ya	Ya	Ya	Ya
5.	Semua kebutuhan di dokumen SRS sudah menggunakan pernyataan yang terukur sehingga dapat diverifikasi.	Ya	Ya	Ya	Ya
6.	Semua istilah, informasi, atau kebutuhan sudah dinyatakan dengan jelas, mudah dipahami, dan tidak ambigu.	Tidak	Tidak	Ya	Ya
7.	Dokumen SRS ini sudah ringkas. Artinya informasi/kebutuhan terkait perangkat lunak sudah dijelaskan	Ya	Ya	Tidak	Tidak

No.	Pertanyaan	Penguji 1		Penguji 2	
		MIL-STD-498	IEEE 830-1998	MIL-STD-498	IEEE 830-1998
	secara singkat namun tetap jelas.				
8.	Dokumen SRS ini sudah lengkap. Artinya semua kebutuhan penting terkait perangkat lunak seperti fungsionalitas, batasan desain dan sebagainya sudah dimuat pada dokumen SRS.	Tidak	Tidak	Tidak Tahu	Tidak Tahu
9.	Dokumen SRS ini sudah ringkas dan lengkap. Artinya semua kebutuhan terkait perangkat lunak sudah dijelaskan secara singkat, jelas, dan tidak ada kebutuhan penting yang terlewat.	Tidak	Tidak	Tidak Tahu	Tidak Tahu
10.	Kedua dokumen SRS ini sudah cukup memenuhi standar penulisan dokumen SRS.	Ya	Ya	Ya	Ya
11.	Subbab Kebutuhan Kemampuan Perangkat Lunak dan Fungsionalitas (subbab 2.3) telah disusun dengan benar.	Ya	Ya	Ya	Ya
12.	Kekurangan apa yang anda temui pada subbab 2.3 di dokumen SRS versi MIL-STD-498?	1. Masih terdapat penjelasan yang membingungkan. 2. Terdapat informasi yang muncul berulang-ulang. 3. Tidak memiliki skenario masukan/keluaran di setiap kemampuan perangkat lunak seperti IEEE. 4. Detil kebutuhan terutama terkait kualitas perangkat lunak masih kurang tepat. 5. Karakteristik data elemen tidak di detailkan. 6. Karakteristik antarmuka tidak lengkap/detil.		1. Tidak memiliki skenario masukan/keluaran di setiap kemampuan perangkat lunak seperti IEEE.	
13.	Kekurangan apa yang anda temui pada	1. Terdapat informasi yang		1. Terdapat informasi yang	

No.	Pertanyaan	Penguji 1		Penguji 2	
		MIL-STD-498	IEEE 830-1998	MIL-STD-498	IEEE 830-1998
	subbab 2.3 di dokumen SRS versi IEEE 830-1998?	muncul berulang-ulang.		tidak penting dan seharusnya tidak perlu dimuat pada dokumen SRS	
14.	Secara umum kedua dokumen ini masih memiliki sejumlah kekurangan, di antaranya:	<ol style="list-style-type: none"> 1. Tidak tersedia notasi diagram seperti <i>usecase</i> diagram, UML, ERD, dsb. 2. Sejumlah kebutuhan tidak diberi penomoran yang unik, sehingga sulit dilakukan pengujian. 3. Kebutuhan yang sama muncul secara berulang-ulang. 4. Detil standar pengembangan di organisasi kurang dieksplor. Misal, Lingkungan tempat sistem akan dideploy, batasan utilitas <i>hardware</i>, Standar operasi autentifikasi pengguna seperti apa? 	<ol style="list-style-type: none"> 1. Tidak tersedia notasi diagram seperti <i>usecase</i> diagram, UML, ERD, dsb. 2. Beberapa kebutuhan bukan merupakan informasi penting yang harus dimuat pada dokumen. 3. Sejumlah kebutuhan tidak diberi penomoran yang unik, sehingga sulit dilakukan pengujian. 		
15.	Dari kedua dokumen SRS, manakah dokumen yang memberikan informasi lebih rinci serta lebih mudah anda pahami?	IEEE 830-1998	IEEE 830-1998		