

**KOMPARASI KUALITAS STANDAR DOKUMEN *SOFTWARE*
REQUIREMENT SPECIFICATION (SRS):
MIL-STD-498 DAN IEEE 830-1998**



Disusun Oleh:

N a m a : Bimo Prakoso

NIM : 13523078

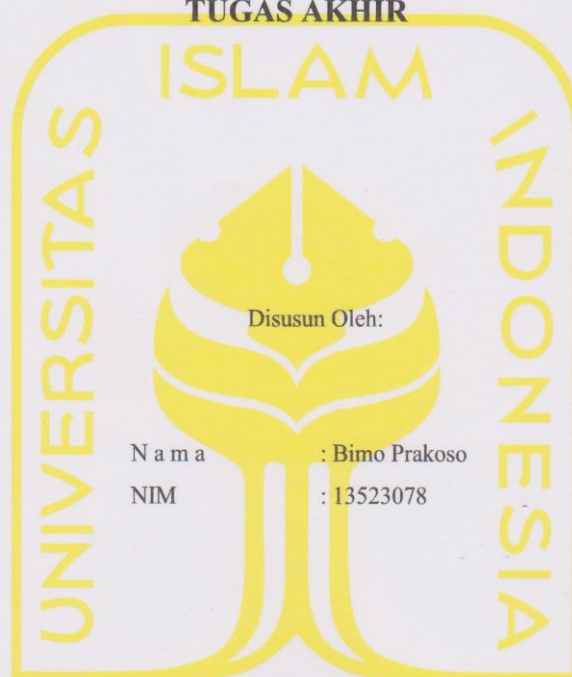
**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**KOMPARASI KUALITAS STANDAR DOKUMEN *SOFTWARE*
REQUIREMENT SPECIFICATION (SRS):
MIL-STD-498 DAN IEEE 830-1998**

TUGAS AKHIR



N a m a : Bimo Prakoso
NIM : 13523078

Yogyakarta, 9 Mei 2018

Pembimbing,

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

HALAMAN PENGESAHAN DOSEN PENGUJI

**KOMPARASI KUALITAS STANDAR DOKUMEN *SOFTWARE*
*REQUIREMENT SPECIFICATION (SRS):*****MIL-STD-498 DAN IEEE 830-1998****TUGAS AKHIR**

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk
memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 9 Mei 2018

Tim Penguji

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

Anggota 1

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

Anggota 2

Sri Mulyati, S.Kom., M.Kom.

Mengetahui,

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 : Bimo Prakoso

NIM : 13523078

Tugas akhir dengan judul:

**KOMPARASI KUALITAS STANDAR DOKUMEN *SOFTWARE*
REQUIREMENT SPECIFICATION (SRS):
MIL-STD-498 DAN IEEE 830-1998**

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, 9 Mei 2018



(Bimo Prakoso)

HALAMAN PERSEMBAHAN

Rasa syukur saya ucapkan yang sebesar-besarnya kepada Allah subhanahu wa'ta'ala atas segala limpahan karunia dan rahmat-Nya sehingga saya dapat menyelesaikan tugas akhir ini dengan baik. Besar harapan agar tugas akhir ini dapat memberikan manfaat kepada saya pribadi maupun pihak lain dikemudian hari.

Saya ingin berterima kasih dan mempersembahkan tugas akhir ini kepada orangtua tercinta, Sumarlan dan Oniq Anjarsari beserta kakak saya Tiara Sari Pertiwi yang tidak pernah lelah memberikan doa, dukungan, serta semangatnya baik selama proses pengerjaan tugas akhir maupun dalam aktivitas sehari-hari. Tak lupa juga saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada teman baik saya Azka Shalihah atas semua semangat dan doa yang telah diberikan.

Tugas akhir ini tidak akan pernah selesai tanpa bantuan berbagai pihak dan kawan. Untuk itu saya berdoa dan berharap agar Allah subhanahu wa'ta'ala selalu memberikan karunia, kekuatan, dan kedamaian untuk semua pihak dan kawan yang telah terlibat dalam pengerjaan tugas akhir ini. Semoga semua yang menjadi harapan kalian dapat disegerakan oleh Allah subhanahu wa'ta'ala.

Bimo Prakoso

Maret 2018

HALAMAN MOTO

“What's done is done. What's gone is gone. One of life's lessons is always moving on.”

— **Roy T. Benneett, *The Light in the Heart***

“Breathe. It's only a bad day, not a bad life.”

— **Johnny Depp**

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji Syukur ke hadirat Allah subhanahu wa'ta'ala, Tuhan semesta alam, yang telah melimpahkan rahmat, taufiq serta hidayat-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Komparasi Kualitas Standar Dokumen *Software Requirement Specification* (SRS): MIL-STD-498 dan IEEE 830-1998 (Studi Kasus: Perangkat Lunak *Job Order Request* PT.Pupuk Kaltim)” dengan baik. Shalawat serta salam tak lupa kita kirimkan kepada Nabi besar Muhammad Shalallahu ‘alaihi wa’salam, karena atas perjuangan beliau kita dapat memperoleh ilmu yang bermanfaat selama ini.

Laporan Tugas Akhir ini dibuat sebagai salah satu syarat memperoleh gelar sarjana Teknik Informatika di Universitas Islam Indonesia serta menjadi sarana bagi penulis untuk mempraktekkan ilmu yang telah diperoleh selama masa studi.

Penulis menyadari jika proses pengerjaan Tugas Akhir ini tidak dapat selesai dengan baik tanpa adanya doa, bimbingan, dorongan, dan bantuan dari berbagai pihak. Oleh sebab itu penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah subhanahu wa'ta'ala, atas rahmat dan hidayahnya penulis dapat menyelesaikan Tugas Akhir ini.
2. Kedua orang tua (Sumarlan, S.E. dan Oniq Anjarsari) dan kakak penulis (Tiara Sari Pertiwi) karena doa dan dukungan yang telah diberikan sehingga penulis dapat menyelesaikan studi di Teknik Informatika dengan baik.
3. Bapak Dr. Imam Djati Widodo, M.Eng. Sc, Selaku Dekan Fakultas Teknologi Industri.
4. Bapak Hendrik, S.T, M.Eng., selaku Ketua Jurusan Teknik Informatika.
5. Bapak Hanson Prihantoro Putro, S.T., M.T., selaku Dosen Pembimbing Tugas Akhir yang telah ikhlas mencurahkan waktu, tenaga, dan saran serta telah mendampingi penulis dalam menyusun Tugas Akhir ini.
6. Ibu Novi Setaini, S.T., M.T., dan Bapak Raden Teduh Dirgahayu S.T., M.Sc., Ph.D., yang sudah mau terlibat pada proses pengujian dokumen SRS yang penulis hasilkan.
7. Ibu Elyza Gustri Wahyuni, S.T., M.Cs., selaku Dosen Pembimbing Akademik.

8. Seluruh Dosen jurusan Teknik Informatika Universitas Islam Indonesia, atas segala ilmu yang pernah diberikan.
9. Kawan-kawan selama masa kuliah Laksa, Miko, Ipan, Vicky, Calex, Anggi, dan kawan lain yang tergabung dalam grup ROAD TO WIMBLEDON.
10. Kawan main, belajar, dan berbicara Azka Shalihah, terima kasih atas waktu, semangat, dan dukungan yang telah diberikan. Semoga Allah subhanahu wa'ta'ala senantiasa memberikan limpahan karunia dan kekuatan kepadamu.
11. Kawan-kawan sekolah Indra, Parit, Agum, Candra, Koirul terima kasih atas dukungan yang telah diberikan.
12. Semua pihak yang telah banyak membantu proses pengerjaan Tugas Akhir dan tidak dapat penulis sebutkan satu persatu.

Penulis amat menyadari jika masih banyak kekurangan pada laporan ini. Oleh karenanya penulis mengharapkan kritik dan saran membangun guna menyempurnakan laporan Tugas Akhir ini. Akhirnya, besar harapan penulis agar laporan ini dapat memiliki manfaat bagi semua pihak, dan atas segala bantuan yang telah diberikan penulis berdoa agar semua pihak yang telah membantu proses pengerjaan Tugas Akhir ini dapat diberikan imbalan yang setimpal dari Allah subhanahu wa'ta'ala, Amin.

Wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, 9 Mei 2018

(Bimo Prakoso)

SARI

Berbagai kebutuhan perangkat lunak yang telah dikumpulkan, dianalisis, kemudian disetujui oleh pihak pelanggan selanjutnya akan dituangkan ke dalam dokumen *Software Requirements Specification* (SRS). Ada berbagai standar yang dapat digunakan untuk membuat dokumen ini seperti IEEE 830-1998, ISO/IEC/IEEE 29148:2011, ataupun MIL-STD-498 (DI-IPSC-81433). Karena standar yang tersedia cukup banyak alhasil penggunaanya seringkali kesulitan untuk menentukan standar yang tepat bagi mereka. Oleh sebab itu penelitian ini dilakukan dengan membandingkan antara standar-standar pembuatan dokumen SRS yang ada untuk mengetahui standar yang dapat menyajikan dokumen SRS dengan kualitas yang ringkas, lengkap, dan mudah diterapkan oleh pengembang pemula. Standar yang dipilih untuk dikomparasi adalah IEEE 830-1998 dan MIL-STD-498 (DI-IPSC-81433) dengan mengangkat studi kasus pengembangan perangkat lunak *Job Order Request* di PT. Pupuk Kaltim.

Dari hasil penelitian yang telah dilakukan diketahui bahwa standar IEEE 830-1998 memiliki lebih banyak keunggulan seperti, dokumen SRS yang dihasilkan lebih lengkap, dokumen panduan dan dokumen SRSnya mudah untuk dipahami, serta mencari referensi terkait standar ini baik yang menggunakan Bahasa Indonesia maupun Bahasa Inggris sangat mudah ditemukan. Sementara MIL-STD-498 memiliki keunggulan dari sisi keringkasannya. Baik keringkasan pada dokumen SRS yang dihasilkan maupun dokumen panduannya. Namun, dari sisi kelengkapan dan kemudahan untuk dimengerti masih menjadi kekurangan pada standar ini. Oleh sebab itu maka standar IEEE 830-1998 ini sangat direkomendasikan untuk digunakan pengembang pemula ketika mereka ingin membuat dokumen SRS.

Kata kunci: Komparasi, MIL-STD-498, IEEE 830-1998, *Job Order Request*.

GLOSARIUM

Dokumen SRS	Merupakan dokumen yang berisi spesifikasi terkait kebutuhan fungsional dan nonfungsional dari perangkat lunak yang akan dikembangkan.
Komparasi	Merupakan proses membandingkan satu objek dengan objek lainnya. objek yang bisa dibandingkan dapat berupa benda, orang, kondisi dan sebagainya.
MIL-STD-498	Merupakan standar militer Amerika Serikat untuk menyeragamkan kebutuhan pengembangan dan dokumentasi perangkat lunak.
IEEE 830-1998	Merupakan <i>best practice</i> keluaran organisasi IEEE dan bertujuan untuk menentukan kebutuhan perangkat lunak yang akan dikembangkan dengan menampilkan sejumlah <i>template</i> dokumen SRS.
<i>Requirement</i>	Kebutuhan (fungsional maupun non-fungsional) dari perangkat lunak atau sistem yang sedang dikembangkan.
<i>Template</i>	<i>Template</i> adalah contoh atau format yang berguna sebagai panduan dalam menciptakan sesuatu.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
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	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian.....	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	5
2.1 <i>Requirement Engineering</i>	5
2.2 Dokumen <i>Software Requirement Specification</i> (SRS).....	9
2.3 Standar MIL-STD-498.....	10
2.4 Standar IEEE 830-1998	11
2.5 Komparasi.....	13
2.6 Penelitian Terkait	16
BAB III METODOLOGI.....	18
3.1 Tahap Melakukan Komparasi	18
3.2 Penentuan <i>Template</i> Dokumen SRS	18
3.3 Pembuatan Dokumen SRS.....	20
3.4 Pengujian Dokumen SRS.....	21

3.5 Mengevaluasi Hasil Komparasi	23
BAB IV HASIL DAN PEMBAHASAN	24
4.1 Pembuatan Dokumen SRS	24
4.1.1 Hasil Dokumen SRS.....	24
4.1.2 Kelebihan dan Kekurangan	30
4.1.3 Kendala Pembuatan Dokumen SRS	33
4.1.4 Rangkuman Perubahan Dokumen SRS	36
4.1.5 Evaluasi Dokumen	39
4.2 Pengujian Dokumen SRS	41
4.3 Evaluasi Hasil Komparasi	44
BAB V KESIMPULAN DAN SARAN	45
5.1 Kesimpulan.....	45
5.2 Saran.....	45
DAFTAR PUSTAKA	47
LAMPIRAN.....	49

DAFTAR TABEL

Tabel 4.1 Hasil Komparasi Berdasarkan Panduan dan Dokumen SRS	24
Tabel 4.2 Daftar Istilah Teknis yang Disesuaikan	34
Tabel 4.3 Rangkuan Perubahan Dokumen SRS format IEEE 830-1998	36
Tabel 4.4 Rangkuan Perubahan Dokumen SRS format MIL-STD-498	38
Tabel 4.5 Daftar Kebutuhan yang Gagal di Implementasikan	39
Tabel 4.6 Daftar Kekurangan Dokumen SRS	41

DAFTAR GAMBAR

Gambar 2.1 Proses <i>Requirements Engineering</i> Menurut Mohammed Javeed Ali	6
Gambar 2.2 Proses <i>Requirements Engineering</i> Menurut Romi Satria Wahono	6
Gambar 3.1 Tahap Komparasi.	18
Gambar 3.2 <i>Template</i> dokumen SRS MIL-STD-498	19
Gambar 3.3 <i>Template</i> dokumen SRS IEEE 830-1998	20
Gambar 4.1 Daftar Isi Dokumen SRS Format MIL-STD-498 Bagian 1	26
Gambar 4.2 Daftar Isi Dokumen SRS Format MIL-STD-498 Bagian 2	27
Gambar 4.3 Daftar Isi Dokumen SRS Format IEEE 830-1998 Bagian 1	28
Gambar 4.4 Daftar Isi Dokumen SRS Format IEEE 830-1998 bagian 2	29
Gambar 4.5 Daftar Isi Dokumen SRS Format IEEE 830-1998 Bagian 3	30

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam proses pengembangan perangkat lunak, pihak analis akan mengumpulkan berbagai kebutuhan perangkat lunak yang diinginkan oleh pelanggan sebelum fase desain dan implementasi berjalan. Proses ini penting dan harus dilakukan dengan benar. Kesalahan pada fase pengumpulan kebutuhan yang terlambat ditangani akan menyebabkan biaya perbaikan yang sangat besar dan proses pengembangan perangkat lunak pun harus diulang kembali dari awal (Graham, Veenendaal, Evans, & Black, 2008). Kebutuhan yang telah dikumpulkan, dianalisis, dan disetujui oleh pelanggan selanjutnya akan dituangkan ke dalam dokumen Spesifikasi Kebutuhan Perangkat Lunak atau dikenal dengan dokumen *Software Requirements Specification* (SRS).

Dokumen SRS berisi penjelasan lengkap mengenai maksud dan tujuan dari sistem yang dikembangkan serta lingkungan di mana produk akan dioperasikan (Rouse, 2016). Dokumen SRS yang ditulis dengan benar akan membantu memenuhi tujuan pembuatan perangkat lunak dan mengurangi resiko kesalahan pada produk yang dihasilkan. Dalam membuat dokumen SRS terdapat berbagai standar yang bisa digunakan sebagai acuan, di antaranya adalah standar keluaran IEEE seri 830 tahun 1993 atau 1998, ISO/IEC/IEEE 29148:2011 yang menjadi standar pengganti dari IEEE 830-1998, ataupun MIL-STD-498 (DI-IPSC-81433) yang dikeluarkan oleh Departemen Pertahanan Amerika Serikat.

Dari semua standar yang ada, IEEE seri 830 merupakan standar yang paling banyak digunakan. Jika kita mengetikkan kata kunci “*SRS document*” pada mesin pencari maka mayoritas panduan maupun contoh dokumen SRS yang muncul akan menggunakan standar IEEE seri 830 ini. Selain itu standar ini juga menjadi rujukan yang digunakan dalam matakuliah Rekayasa Perangkat Lunak di Teknik Informatika Universitas Islam Indonesia (UII). Hal ini membuktikan bahwa meskipun standar pengganti atau versi terbarunya telah tersedia namun IEEE seri 830 masih menjadi standar yang paling sering digunakan dalam membuat dokumen SRS. Sedangkan standar MIL-STD-498 sendiri umumnya digunakan untuk membangun perangkat lunak dengan tingkat kerumitan maupun bahaya yang tinggi dan biasanya standar ini digunakan untuk kebutuhan pengembangan perangkat lunak di bidang militer.

Banyaknya standar yang tersedia sudah seharusnya membuat proses pembuatan dokumen SRS menjadi lebih mudah. Akan tetapi, banyaknya pilihan akan standar yang ada seringkali malah membuat penggunaanya (dalam hal ini pengembang pemula) kesulitan dalam menentukan standar yang tepat bagi mereka. Oleh sebab itu, penelitian ini akan membandingkan standar-standar pembuatan dokumen SRS yang ada untuk mengetahui manakah standar yang dapat menyajikan dokumen SRS dengan kualitas yang ringkas, lengkap, dan mudah diterapkan oleh pengembang pemula. Standar yang dipilih untuk dikomparasi adalah IEEE 830-1998 dan MIL-STD-498 (DI-IPSC-81433) dengan mengangkat studi kasus pengembangan perangkat lunak *Job Order Request* di PT. Pupuk Kaltim, Kalimantan Timur.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah dijelaskan pada latar belakang di atas, maka rumusan masalah dari penelitian ini adalah bagaimana perbandingan standar MIL-STD-498 dan IEEE 830-1998 dalam menyajikan sebuah dokumen SRS dengan kualitas yang ringkas, lengkap, dan mudah diterapkan oleh pengembang pemula.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

- a. Format penulisan dokumen SRS menggunakan standar Pengembangan Perangkat Lunak dan Dokumentasi yang dikeluarkan Departemen Pertahanan Amerika Serikat, MIL-STD-498 dan IEEE 830-1998 *Recommended Practice for Software Requirements Specifications*.
- b. Kualitas artinya, standar pembuatan dokumen SRS yang dapat menyajikan dokumen secara ringkas namun tetap lengkap, dan mudah digunakan oleh pengembang pemula (Mahasiswa Teknik Informatika UII). Kriteria ringkas dan lengkap akan dikaji kemudian.
- c. Penerapan standar MIL-STD-498 dan IEEE 830-1998 dilakukan pada proses pengembangan perangkat lunak *Job Order Request* (JOR) PT. Pupuk Kaltim.
- d. Hasil akhir penelitian ini berupa rekomendasi yang ditujukan kepada pengembang pemula, khususnya mahasiswa Teknik Informatika UII mengenai standar yang mudah digunakan dalam membuat dokumen SRS sehingga dokumen yang dihasilkan akan memenuhi kualitas ringkas namun tetap lengkap.

- e. Setiap perubahan spesifikasi maupun kebutuhan pada dokumen SRS akan diikuti dengan perubahan pada sistem.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah melakukan kegiatan komparasi terhadap dokumen SRS yang dihasilkan dari standar MIL-STD-498 dan IEEE 830-1998 untuk mencari tahu dan memberi rekomendasi kepada pengembang pemula mengenai standar pembuatan dokumen SRS yang dapat memenuhi kualitas ringkas namun tetap lengkap dan mudah untuk mereka gunakan.

1.5 Manfaat Penelitian

Manfaat dari dilakukannya penelitian berikut adalah:

- a. Memberikan informasi perihal kelebihan dan kekurangan dari standar yang dibandingkan.
- b. Memberikan informasi tentang standar mana yang menghasilkan dokumen SRS dengan kualitas ringkas, lengkap, dan mudah diaplikasikan oleh pengembang pemula.
- c. Dengan standar yang tepat pengembang pemula dapat mengurangi resiko kesalahan dalam mengembangkan perangkat lunak.

1.6 Metodologi Penelitian

- a. Studi Kepustakaan

Pada tahap ini akan dikumpulkan berbagai informasi terkait dokumen SRS, standar MIL-STD-498, IEEE 830-1998, dan penelitian yang berkaitan dengan cara penilaian kualitas suatu dokumentasi.

- b. Penentuan Parameter Komparasi

Parameter komparasi yang digunakan adalah ringkas dan lengkap. Dokumen SRS dapat dikatakan ringkas dan lengkap apabila dokumen yang dihasilkan tidak terlalu panjang namun semua informasi yang dibutuhkan untuk mengembangkan perangkat lunak telah tersedia.

- c. Implementasi Standar Dokumen

Setelah informasi yang dikumpulkan dirasa cukup dan parameter komparasi telah ditentukan, selanjutnya penelitian dimulai dengan membuat dokumen SRS menggunakan kedua standar yang telah ditentukan yaitu MIL-STD-498 dan IEEE 830-

1998. Apabila diperlukan nanti akan ditambahkan *prototype* spesifikasi desain perangkat lunak, dan *prototype* programnya.

d. Perbaikan Dokumen SRS dan Perangkat Lunak

Jika dokumen SRS harus diperbaiki atau dimodifikasi sesuai rekomendasi dosen ahli rekayasa perangkat lunak Universitas Islam Indonesia, baik dari segi *requirements*, desain, dan sebagainya, maka segala perubahan pada dokumen tersebut akan diikuti perubahan dari sisi perangkat lunaknya.

e. Pengujian Hasil Komparasi

Pada tahap ini kegiatan komparasi telah selesai dilakukan, dan akan diuji untuk mengukur kebenaran hasil komparasi dengan berkonsultasi kepada dosen ahli rekayasa perangkat lunak dan mahasiswa Teknik Informatika (sebagai desainer perangkat lunak) di Universitas Islam Indonesia.

1.7 Sistematika Penulisan

Untuk memudahkan pembaca dalam memahami bagian-bagian yang terkandung dalam laporan ini, maka penjelasan mengenai sistematika penulisan laporan dapat dilihat seperti berikut:

Bab I Pendahuluan, berisi penjelasan terkait masalah yang diangkat pada penelitian ini dan terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II Landasan Teori, menjelaskan tentang teori-teori dasar dan berbagai rujukan terkait penelitian yang dilakukan, seperti konsep dasar *requirements engineering*, dokumen SRS, IEEE 830-1998, dan MIL-STD-498.

Bab III Metodologi, menjelaskan aspek kualitas dari dokumen SRS yang akan dibandingkan, serta langkah-langkah atau skenario ketika proses komparasi dilakukan.

Bab IV Hasil dan Pembahasan, berisi hasil komparasi yang telah dilakukan beserta rekomendasi mengenai standar yang tepat untuk digunakan oleh pengembang pemula dalam membuat dokumen SRS.

Bab V Kesimpulan dan Saran, kesimpulan menjelaskan hasil keseluruhan yang didapatkan dari proses penelitian, sedangkan saran berisi masukan untuk penelitian selanjutnya sehingga dapat ditingkatkan menjadi lebih baik lagi.

BAB II

LANDASAN TEORI

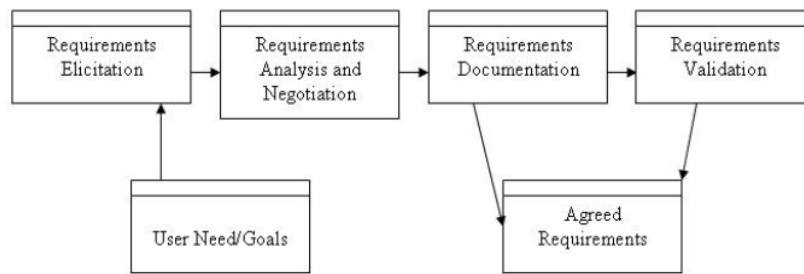
2.1 *Requirement Engineering*

Setiap sistem maupun perangkat lunak memiliki *requirement* yang berbeda-beda. *Requirement* adalah hal-hal yang harus dilakukan oleh sistem (kebutuhan fungsional), kualitas yang harus dipenuhi oleh sistem (kebutuhan non-fungsional), dan berbagai batasan yang dimiliki pengembang saat membuat sistem (Westfall, 2006). *Requirement* harus dibuat dan disetujui oleh pelanggan, pengguna, dan pengembang sebelum perangkat lunak dikembangkan (Westfall, 2006).

Menurut (Wahono, 2006) *requirement engineering* merupakan tahap awal dalam rekayasa perangkat lunak untuk mengumpulkan, memahami, lalu menetapkan setiap kebutuhan dari pengguna maupun pelanggan. *Requirement engineering* menurut (Ali, 2006) adalah proses menemukan *requirement* dari pihak yang membutuhkan sistem kemudian mendokumentasikan kebutuhan-kebutuhan tersebut untuk dianalisis, dikomunikasikan, dan diimplementasikan.

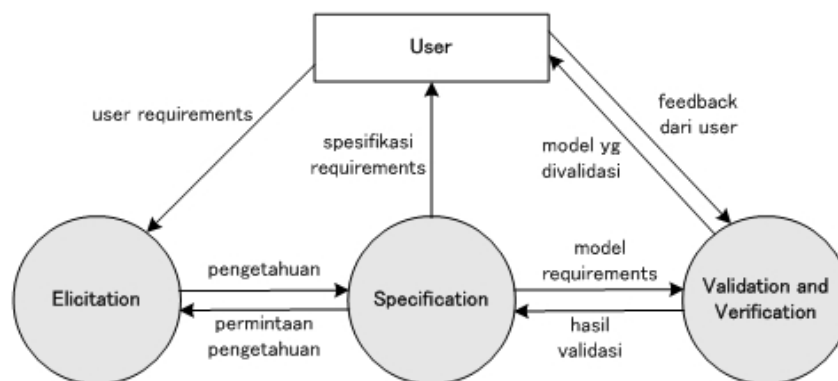
Proses mencari dan mengumpulkan kebutuhan ini merupakan tahap rumit dan beresiko. Kesalahan yang kerap ditemui umumnya karena ketidakmampuan pengembang dalam memahami kebutuhan pelanggan, sehingga terus terjadi perubahan pada produk yang dihasilkan. Kondisi tersebut sama seperti istilah “*the rock problem*” yang menggambarkan perasaan frustrasi pengembang karena perubahan *requirement* yang terus menerus terjadi dari pihak pelanggan. Disisi lain pelanggan pada akhirnya juga akan merasa tertekan karena kebutuhan yang mereka sampaikan tidak dapat dipahami dengan jelas oleh pengembang (Leffingwell & Widrig, 2003).

Dari beberapa referensi yang ditemukan, *requirements engineering* memiliki proses yang berbeda-beda. Menurut (Wahono, 2006) *requirements engineering* memiliki 3 proses utama yaitu *requirement elicitation*, *requirement spesification*, dan *requirement validation* dan *verification*. Menurut (Pressman, 2009) ada 5 proses pada *requirement engineering* yaitu *requirement elicitation*, *requirement analysis* dan *negotiation*, *requirement spesification*, *system modeling*, *requirement validation*, dan *requirement managemen*. Sedangkan (Ali, 2006) mengatakan *requirement engineering* terdiri dari *requirements elicitation*, *requirements analysis*, *requirements negotiation* dan *validation*.



Gambar 2.1 Proses *Requirements Engineering* Menurut Mohammed Javeed Ali

Sumber: Ali (2006)



Gambar 2.2 Proses *Requirements Engineering* Menurut Romi Satria Wahono

Sumber : Wahono (2006)

a. *Requirement Elicitation*

Requirement elicitation merupakan proses mengumpulkan kebutuhan dari sistem yang akan dibuat. *Elicitation* menjadi tahap awal pada proses *requirement engineering* dan dikerjakan oleh semua pihak yang terlibat pada proses pengembangan perangkat lunak. Beberapa masalah yang kerap dijumpai pada proses *requirement elicitation* salah satunya adalah keterbatasan pengetahuan pengembang mengenai sistem yang dibutuhkan oleh pelanggan (*gap knowledge*) (Wahono, 2006). Untuk menghindari masalah tersebut maka komunikasi harus dilakukan terus menerus antara pengembang dengan pelanggan. Proses komunikasi ini kemudian melahirkan beberapa metode pada tahap *elicitation*. Menurut (Ganesh & Gunda, 2008) ada dua tipe proses *elicitation*, yaitu:

1. *Classic Requirements Elicitation Techniques: Interview*, kuisisioner, observasi.

2. *Modern Requirements Elicitation Techniques*: Prototipe, *requirements reuse*, *skenario*, *brainstroming*, *Joint Application Development*, *user centered design*.

b. *Requirements Analysis*

Requirement analysis berguna untuk memecahkan masalah dan mencapai kesepakatan tentang perubahan kebutuhan pada sistem maupun perangkat lunak (Ali, 2006). Pada proses ini ada beberapa pertanyaan yang harus mendapatkan jawaban secara jelas, diantaranya adalah (Pressman, 2009):

1. Apakah setiap kebutuhan yang telah dikumpulkan konsisten dengan tujuan dibuatnya sistem?
2. Apakah kebutuhan dari pelanggan benar-benar mereka butuhkan atau kebutuhan itu hanya berupa fitur yang tidak begitu penting?
3. Apakah suatu kebutuhan memiliki konflik dengan kebutuhan lainnya?
4. Apakah kebutuhan yang diperoleh dapat diuji ketika telah di implementasikan?

Seringkali pelanggan mengajukan *requirement* berbeda-beda dan saling bertentangan antara satu dan lainnya. Untuk menghindari hal tersebut seorang analis harus mengajak semua orang yang terlibat dalam pengembangan produk untuk sama-sama menentukan prioritas dari kebutuhan yang paling mereka perlukan. Dengan menggunakan pendekatan iteratif, nantinya setiap kebutuhan akan dieliminasi, digabung, dan/atau dimodifikasi sehingga setiap pihak yang terlibat pada pengembangan produk dapat mencapai level kepuasan yang sama (Pressman, 2009).

c. *Requirements Specification*

Spesifikasi perangkat lunak dapat ditulis ke dalam berbagai bentuk seperti dokumen, grafik, model formal matematika, prototipe, skenario, atau kombinasi dari semuanya (Pressman, 2009). Spesifikasi ini menjadi dasar bagi *programmer*, *database desainer*, dan *hardware enginer* dalam membuat program yang dibutuhkan oleh pelanggan (Pressman, 2009). Pendekatan paling baik untuk membuat spesifikasi pada sistem berskala besar yaitu menggunakan dokumen yang dilengkapi deskripsi memakai bahasa sehari-hari lalu ditambahkan dengan berbagai grafik seperti DFD, ERD, UML, dan sebagainya untuk memudahkan pengembang dalam proses pembuatan produk. Namun untuk sistem dengan skala yang lebih kecil umumnya mendeskripsikan semua fitur yang ada menggunakan skenario akan lebih mudah untuk dipahami (Pressman, 2009). Menggunakan *template* dokumen spesifikasi dari standar tertentu juga akan membantu proses pembuatan spesifikasi produk sehingga kebutuhan dapat ditulis secara konsisten

dan membantu pembaca dalam memahami isi dari dokumen spesifikasi tersebut (Pressman, 2009).

d. *System Modeling*

Untuk mencegah terjadinya masalah pada proses pengembangan perangkat lunak maka harus dilakukan aktifitas pemodelan sistem sebelum kegiatan pengembangan dilakukan. Hal ini akan membantu pengembang memprediksi tingkat keberhasilan maupun kegagalan dari sistem yang dikembangkan tanpa harus membuatnya terlebih dulu (Hedwing, 2011). *System modeling* atau pemodelan sistem adalah gambaran sederhana dari sistem yang dikembangkan untuk membantu semua pihak yang terlibat dalam pengembangan perangkat lunak memahami, memprediksi, dan mengendalikan sifat dari sistem yang dibuat (Hedwing, 2011). Proses pemodelan sistem akan terus dilakukan jika model yang dibuat dirasa belum memberikan solusi yang memuaskan. Namun jika dirasa hasilnya telah sesuai keinginan maka kegiatan pemodelan dapat dihentikan kemudian model tersebut dapat diaplikasikan secara nyata (Hedwing, 2011).

e. *Requirement Validation*

Spesifikasi sistem dan seluruh informasi yang diperoleh dari proses *requirement engineering* akan dinilai kualitasnya pada tahap *requirement validation*. Tahap ini akan memastikan bahwa dokumen spesifikasi atau SRS yang dihasilkan itu lengkap, konsisten, layak, dibutuhkan, tidak ambigu, dapat diverifikasi dan semua kesalahan yang muncul dapat segera ditemukan dan diperbaiki (Requirement Management School, 2010). Proses ini sangat penting mengingat kesalahan pada fase analisis kebutuhan yang dibiarkan berlarut-larut akan memberikan kerugian sangat besar bagi pengembang maupun pelanggan, sehingga untuk mengurangi resiko tersebut semua pemangku kepentingan baik pengembang, pelanggan, pengguna, dan pihak lainnya akan berkumpul untuk memeriksa kembali spesifikasi dari sistem yang telah dibuat, mencari kesalahan pada *requirement*, menemukan informasi yang hilang, *requirement* yang memiliki konflik dengan *requirement* lain, atau juga adanya *requirement* yang tidak realistis (Pressman, 2009). Ada beberapa teknik yang dapat digunakan untuk melakukan validasi terhadap *requirement*, seperti teknik *review* dan inspeksi, prototipe, *traceability*, dan pengujian (Requirement Management School, 2010).

2.2 Dokumen *Software Requirement Specification* (SRS)

Dokumen SRS dihasilkan pada tahap analisis kebutuhan. Dokumen ini dapat ditulis oleh pengguna, pelanggan atau pengembang (analisis) (Ali, 2006). Dokumen SRS menjadi alat komunikasi antara pihak pelanggan dan pengembang agar masing-masing pihak dapat mengetahui secara pasti kebutuhan perangkat lunak yang akan dikembangkan dan bisa memvalidasi kebenarannya. Nantinya, apabila dokumen SRS telah disetujui oleh pihak pelanggan dan pengembang, selanjutnya dokumen ini akan diserahkan kepada desainer sebagai panduan dalam merancang perangkat lunak yang dibutuhkan oleh pelanggan. Terdapat berbagai standar yang bisa digunakan sebagai acuan dalam membuat dokumen SRS, seperti:

- a. IEEE 830-1993 *Recommended Practice for Software Requirements Specifications*
- b. IEEE 830-1998 *Recommended Practice for Software Requirements Specifications*
- c. MIL-STD-498 (DI-IPSC-81433)
- d. ISO/IEC/IEEE 29148:2011

Pembuatan dokumen SRS harus dilakukan secara benar, sebab kesalahan pada dokumen SRS yang baru diketahui pada fase desain, implementasi, maupun pengujian akan memakan biaya perbaikan yang sangat besar. Kesalahan karena kurangnya pengetahuan atau pengalaman analisis (*knowledge error*) tentang sistem yang dikembangkan menjadi kesalahan yang kerap dijumpai saat membuat dokumen SRS (Ali, 2006).

Dokumen ini berisi fungsionalitas dan non-fungsionalitas dari sistem yang dikembangkan (Vie, 2007). Fungsionalitas terkait hal-hal yang harus dapat dikerjakan oleh sistem, dan non-fungsionalitas adalah kualitas yang harus dipenuhi oleh sistem seperti keamanan, performa, efisiensi, keandalan, dan perawatan. Dokumen SRS dapat dikatakan memiliki kualitas yang baik apabila memenuhi beberapa karakteristik berikut ini (Ali, 2006):

- a. Tidak Ambigu

Dokumen SRS dikatakan tidak ambigu jika dan hanya jika setiap *requirement* yang dimuat di dalamnya hanya memiliki satu interpretasi saja.

- b. Benar

Dokumen SRS dikatakan benar apabila setiap *requirement* yang ada di dalamnya sesuai dengan yang dibutuhkan pelanggan.

- c. Lengkap

Dokumen SRS dikatakan lengkap jika semua yang harus dilakukan perangkat lunak termuat di dalam dokumen, semua halaman, tabel, gambar diberi angka, dan sumber

referensi dalam membuat dokumen dihadirkan. Tidak boleh ada *requirement* yang belum ditentukan atau ambigu.

d. Mudah Dipahami

Dokumen SRS harus mudah dipahami oleh semua pihak yang terlibat dalam aktivitas pengembangan perangkat lunak seperti pengguna, pelanggan, maupun pengembang.

e. Dapat Diverifikasi

Dokumen SRS dapat diverifikasi jika dan hanya jika terdapat proses atau *requirement* yang dapat diperiksa oleh manusia atau mesin sehingga bisa dipastikan bahwa *requirement* maupun proses tersebut telah sesuai dengan kebutuhan pelanggan.

f. *Traceable*

Dokumen SRS yang baik harus mudah ditelusuri sehingga memudahkan pengembang dalam mencari *requirement* yang dibutuhkan pelanggan, membantu proses pengujian dan membantu pembaca mencari informasi yang ia butuhkan dalam dokumen.

g. Mudah Dimodifikasi

Karena proses pembuatan perangkat lunak dapat berubah tanpa diduga, maka perubahan juga harus dilakukan pada dokumen SRS-nya. Dokumen SRS harus dapat dimodifikasi dengan mudah ketika dibutuhkan dan perubahan yang dilakukan harus menjamin konsistensi dari dokumen ini.

h. Tidak Boleh Ada Perulangan

Informasi yang tersimpan lebih dari satu bagian pada dokumen SRS mungkin akan memudahkan suatu pihak dalam membaca dokumen ini namun ia juga berpotensi menjadi penyebab masalah ketika informasi pada suatu bagian diubah namun informasi yang sama pada bagian lain tidak diubah sehingga menyebabkan dokumen menjadi ambigu.

i. Ringkas

Dokumen SRS sebaiknya tidak dibuat terlalu panjang namun semua informasi yang dibutuhkan oleh semua pihak harus tetap tersedia dengan lengkap di dalamnya.

2.3 Standar MIL-STD-498

Departemen Pertahanan Amerika Serikat pada tanggal 8 November 1994 telah menyetujui penggunaan standar baru terkait pengembangan perangkat lunak dan dokumentasi yang diberi nama MIL-STD-498. Standar tersebut dihasilkan dari pertemuan yang dilakukan setiap 2 bulan sekali selama 2,5 tahun oleh perkumpulan bernama *Harmonization Working*

Group (HWG) pada Oktober, 1991 dan terdiri dari seluruh perwakilan Departemen Pertahanan Amerika Serikat seperti Angkatan Darat, Angkatan Laut, serta Angkatan Udara dan agen-agen federal dari negara lain seperti Menteri Pertahanan Jerman, Kanada, dan Britania Raya (Departemen of Defense, 1994).

Ada 4 tujuan utama dari munculnya standar ini yaitu : [1] menggabungkan standar yang telah ada sebelumnya yakni standar DOD-STD-2167A yang digunakan sebagai standar sistem persenjataan dan DOD-STD-7935A yang menjadi standar sistem informasi otomatis. [2] memperbaiki masalah-masalah yang muncul dari standar-standar sebelumnya. [3] memastikan kesesuaian dengan arahan, instruksi, dan standar lain yang ada pada Departemen Pertahan. [4] menyediakan basis untuk Amerika Serikat dalam mengimplementasikan standar ISO/IEC 12207, *Software Life Cycle Process* (Departemen of Defense, 1994).

Tidak seperti pendahulunya, standar MIL-STD-498 membebaskan pengembang memilih model pengembangan yang cocok bagi proyek yang mereka kerjakan. Selain itu standar ini juga dapat digunakan secara gratis tanpa perlu membayar apapun. Kelebihan lain yang dimiliki oleh MIL-STD-498 adalah standar ini telah mencakup semua kebutuhan dokumentasi selama proses pengembangan perangkat lunak, sehingga kita tidak perlu menggunakan atau memanggil standar lain ketika memakai MIL-STD-498. Hal ini cukup berbeda dengan beberapa standar lainnya yang terkadang mengharuskan kita untuk menambah penggunaan standar berbeda dan seringkali standar yang dibutuhkan tidak tersedia secara gratis.

2.4 Standar IEEE 830-1998

IEEE 830-1998 *Recommended Practice for Software Requirements Specifications* adalah standar yang dikeluarkan organisasi internasional bernama IEEE dan telah mendapat persetujuan untuk digunakan pada tanggal 25 juni 1998. IEEE 830-1998 berisi penjelasan mengenai bagaimana cara membuat dokumen SRS yang baik dan seperti apa dokumen yang memiliki kualitas bagus dengan menghadirkan *template* penulisan dokumennya (IEEE Computer Society, 1998).

IEEE 830-1998 merupakan panduan yang sering digunakan ketika membuat dokumen SRS. Standar ini adalah hasil revisi dari standar yang ada sebelumnya yaitu IEEE 830-1993. Karena bukan domain milik publik sehingga untuk dapat menggunakan standar ini kita wajib membelinya terlebih dahulu.

Dokumen panduan IEEE 830-1998 terbagi ke dalam lima bagian. Bagian pertama menjelaskan ruang lingkup dari standar IEEE 830-1998. Bagian kedua berisi daftar referensi. Bagian ketiga menyediakan definisi terkait istilah spesifik yang digunakan dokumen ini. Bagian keempat menyediakan informasi dasar mengenai cara menulis dokumen SRS yang baik. Lalu di bagian terakhir menjelaskan elemen-elemen penting yang ada pada dokumen SRS. IEEE 830-1998 juga dilengkapi dengan dua buah lampiran. Lampiran pertama berisi alternatif *template* dokumen SRS dan bagian kedua menyediakan panduan tentang kesesuaian dengan standar IEEE/EIA 12207.1-1997 (IEEE Computer Society, 1998).

2.5 PT. Pupuk Kaltim

PT. Pupuk Kaltim merupakan salah satu anak perusahaan PT. Pupuk Indonesia Holding Company (PIHC) yang bergerak di sektor industri pupuk dan kimia. Berlokasi di kota Bontang, Kalimantan Timur, PT. Pupuk Kaltim menjadi perusahaan penghasil urea dan amoniak terbesar di Indonesia dengan kapasitas produksi mencapai 2,98 juta ton urea dan 1,85 juta ton amoniak per tahun. PT. Pupuk Kaltim bertujuan untuk memenuhi kebutuhan pupuk domestik pada sektor tanaman pangan melalui distribusi pupuk bersubsidi ke wilayah pemasaran Indonesia Timur dan sektor tanaman perkebunan serta industri untuk produk nonsubsidi ke seluruh wilayah Indonesia dan ekspor.

Dengan 8 unit pabrik yang beroperasi PT. Pupuk Kaltim memiliki sekitar 2.400 tenaga kerja organik dan 1.200 tenaga kerja *outsourcing* dari berbagai daerah di Indonesia (Nugraha & Afandi, 2016). Untuk menunjang seluruh kebutuhan karyawannya, PT. Pupuk Kaltim menyediakan layanan yang memungkinkan seluruh karyawan organik untuk mengajukan berbagai kebutuhan yang mereka perlukan kepada perusahaan, seperti kebutuhan peminjaman perlengkapan acara, perbaikan AC, listrik, dan sebagainya. Untuk mengetahui lebih jelas terkait layanan tersebut dapat dilihat pada subbab 2.6.

2.6 Job Order Request (JOR)

Job Order Request (JOR) merupakan “permintaan pelayanan dengan mengirimkan pekerja untuk menyelesaikan suatu permasalahan” (Nugraha & Afandi, 2016). Dengan adanya JOR ini setiap karyawan di PT. Pupuk Kaltim dapat mengajukan permintaan layanan yang mereka butuhkan kepada perusahaan melalui Departemen Pelayanan Umum (DPU) selaku pengelola JOR. Nantinya, seluruh permintaan yang telah diajukan tersebut akan

ditampung oleh DPU dan dikirimkan ke instansi-instansi yang bertugas untuk memenuhi atau menyelesaikan permintaan tersebut. Beberapa layanan yang disediakan oleh JOR yaitu:

- a. Peminjaman perlengkapan acara seperti, panggung, tarub, kursi lipat, umbul-umbul, meja tamu, podium, karpet, dll.
- b. Peminjaman sarana pertemuan seperti, ruang Mahoni, ruang Meranti, ruang Rajawali, gedung Koperasi, dan gedung kantin.
- c. Perbaikan listrik, AC, sipil, dan *plumbing* pada gedung, area perkantoran dan rumah dinas karyawan PT. Pupuk Kaltim.
- d. Permintaan konsumsi seperti nasi kotak atau snek.
- e. Permintaan akomodasi untuk tempat menginap tamu perusahaan seperti hotel Equator, hotel Sintuk, wisma tamu Equator, President suite, wisma Seruni, dll.
- f. Serta barang-barang habis pakai (*consumable*) seperti, Aqua galon, kopi, teh, gula, tisu, dll.

Layanan JOR hanya bisa diajukan oleh karyawan organik PT. Pupuk Kaltim. Untuk dapat mengajukan layanan ini mereka harus mengisi *form* yang disediakan oleh DPU kemudian menunggu hingga *form* tersebut mendapat persetujuan dari pihak otorisator seperti Kepala seksi DPU, Superintendant DPU, dan Manager DPU.

2.7 Komparasi

Komparasi adalah metode untuk memeriksa kesamaan dan/atau perbedaan antara dua buah objek seperti orang, tempat, gagasan, maupun benda (Nordquist, 2017). Komparasi dapat dilakukan dengan berkonsultasi atau melakukan wawancara kepada pihak ahli, maupun pengguna secara umum. Selain kedua pihak tersebut aktifitas komparasi juga dapat dilakukan dengan memanfaatkan teknik tertentu seperti pengujian *usability*. *Usability* adalah proses memastikan suatu produk dapat bekerja dengan baik entah produk itu adalah halaman web, pesawat tempur, atau sebuah pintu, namun penggunaanya dapat dengan mudah memakai produk tersebut tanpa harus merasa frustrasi (Krug, 2000). Pengujian *usability* terhadap suatu produk penting untuk dilakukan, termasuk pada dokumentasi teknis. Sebab banyaknya dokumen teknis yang membingungkan dan tidak intuitif mengakibatkan pembaca enggan mencari informasi yang mereka butuhkan di dalam dokumen dan memilih untuk langsung menanyakan masalah yang mereka hadapi kepada pihak lain (Collins, 2010). Untuk melakukan komparasi menggunakan pengujian *usability*, kita dapat menggunakan beragam skenario maupun metrik yang sesuai dengan tujuan dilakukannya pengujian. (Tullis dan

albert, 2008) merangkum 10 skenario yang kerap digunakan saat melakukan pengujian *usability*, seperti:

a. *Completing Transaction*

Skenario ini bertujuan untuk memastikan proses transaksi pada suatu produk dapat berjalan sebaik mungkin.

b. *Comparing Product*

Skenario ini berguna untuk membandingkan suatu produk dengan produk yang lain sehingga kekuatan serta kelemahan yang dimiliki produk tersebut dapat diketahui.

c. *Evaluating Frequent Use of the Same Product*

Skenario ini bertujuan untuk memastikan bahwa produk-produk tersebut dapat digunakan dengan mudah dan sangat efisien.

d. *Evaluating Navigation and /or Information Architecture*

Skenario ini bertujuan agar pengguna bisa menemukan informasi yang mereka inginkan dengan cepat dan mudah, mereka bisa secara leluasa bergerak antara satu halaman dengan halaman lain dan mengetahui berbagai pilihan yang dapat mereka gunakan.

e. *Increasing Awareness*

Fokus pada skenario ini lebih kepada peningkatan kesadaran konsumen terhadap produk yang kita miliki.

f. *Problem Discovery*

Tujuan dari *Problem Discovery* adalah mencari tahu masalah signifikan yang terdapat pada suatu produk.

g. *Maximizing Usability for a Critical Product*

Produk penting (*critical product*) wajib untuk digunakan secara mudah dan efisien oleh setiap penggunanya, karena dampak negatif yang akan didapat apabila suatu pengguna gagal mengaplikasikan produk tersebut.

h. *Creating an Overall Positive User Experience*

Fokus dari skenario ini adalah mencari tahu persepsi setiap pengguna ketika memakai produk yang diuji.

i. *Evaluating the Impact of Subtle Changes*

Skenario ini bertujuan untuk mencari tahu dampak yang dirasakan pengguna terhadap perubahan desain yang telah dilakukan.

j. *Comparing Alternative Designs*

Sama seperti tujuan dari skenario *comparing product* namun objek yang dibandingkan berupa *prototype* dari beberapa alternatif desain. Pengujian ini dilakukan sebelum desain produk dikembangkan.

Adapun metrik yang dapat digunakan untuk melakukan pengujian *usability*, adalah seperti berikut:

a. *Task Success*

Task success merupakan metrik yang mengukur efektivitas pengguna dalam menyelesaikan suatu tugas. Ada dua cara untuk mengukur *task succes* yaitu *binary success* dan *level of success*.

b. *Time-on-Task*

Time-on-task atau *completion time* atau *task time* merupakan metrik untuk menghitung durasi waktu yang dibutuhkan setiap responden guna menyelesaikan setiap tugas.

c. *Errors*

Metrik ini berguna jika kita ingin mengetahui tindakan yang dapat menyebabkan terjadinya suatu kegagalan, jumlah kegagalan yang dibuat, letak kegagalan yang dilakukan responden pada suatu produk dan sebagainya.

d. *Efficiency*

Efficiency digunakan untuk menghitung upaya yang dibutuhkan seorang partisipan dalam menyelesaikan tugas. Beberapa cara untuk mengukur *efficiency* yakni: Lostness dan Kombinasi Task Success dan time-on-task.

e. *Learnability*

Learnability mengukur tingkat kemudahan suatu produk untuk dipelajari.

f. *Issues-Based Metrics*

Metrik ini umumnya berbentuk kualitatif. Data yang diambil berupa berbagai masalah yang dialami peserta selama menggunakan produk dan penyebab masalah tersebut.

g. *Self-Report Metrics*

Ukuran ini digunakan untuk menanyakan perasaan atau persepsi pengguna ketika menggunakan produk yang diuji.

h. *Metrik Behavioral dan physiological*

Metrik ini digunakan untuk mengobservasi tingkah laku pengguna sehingga kita memiliki gambaran lain terkait produk yang diteliti.

i. *Combined and Comparative Metrics*

Teknik ini menggabungkan beberapa metrik menjadi satu atau membandingkan data *usability* yang telah diperoleh kepada ahli atau dengan hasil yang ideal untuk melahirkan metrik baru

j. *Live Website Data*

Metrik ini memaksa kita mempelajari aktivitas yang dilakukan responden pada suatu halaman *web*, seperti mencari tahu halaman yang mereka kunjungi, tautan yang mereka klik, dan berbagai informasi lainnya.

k. *Card-Sorting Data*

Card-Sorting Data adalah metode untuk membantu mengevaluasi arsitektur informasi pada suatu halaman web.

2.8 Penelitian Terkait

Hingga kini penulis belum dapat menemukan penelitian yang membandingkan kualitas dokumen SRS dari satu standar dengan standar lainnya, khususnya MIL-STD-498 dan IEEE 830-1998. Penelitian yang kerap ditemui umumnya tidak membandingkan kualitas antara satu dokumen dengan dokumen yang lain, melainkan hanya melakukan penilaian kualitas dari satu buah dokumen saja.

Seperti paper yang dihasilkan oleh (Arthur & Stevens, 1992) di mana mereka menilai kualitas satu buah dokumen SRS dengan mengidentifikasi struktur taksonomi yang berhubungan dengan karakteristik dokumen tersebut. Kemudian (Ali, 2006), walaupun tesisnya bertemakan pengukuran kualitas dokumen SRS, namun ia lebih menekankan pemanfaatan perangkat lunak untuk mengukur kualitas suatu dokumentasi secara otomatis (*automated requirements measurement tool*), contohnya *requirement IBM Rational RequisitePro*, *Dynamic Object Oriented Requirements System (DOORS)*, dan *Requirements Use Case Tool*.

Penelitian yang dilakukan kali ini cukup berbeda dengan kedua penelitian tersebut. Sebab penilaian kualitas tidak hanya dilakukan pada satu buah dokumen SRS saja, melainkan kepada 2 buah dokumen SRS beserta standarnya. Kriteria kualitas nantinya akan mengacu pada beberapa karakteristik dokumen SRS yang baik seperti kebenaran, kelengkapan, keringkasan, dan lainnya (Ali, 2006). Kriteria-kriteria tersebut akan digunakan sebagai tolok ukur dalam melakukan pengujian dokumen kepada pihak ahli maupun pengujian *usability*.

Langkah-langkah yang harus dipersiapkan untuk melakukan pengujian *usability* nantinya akan mengacu pada tesis yang dihasilkan oleh (Collins, 2010). Ia menekankan pentingnya melakukan pengujian *usability* terhadap suatu dokumen teknis untuk memperbaiki kualitas dokumen yang dihasilkan dan memastikan suatu produk maupun dokumentasi dapat benar-benar digunakan oleh penggunanya.

BAB III METODOLOGI

3.1 Tahap Melakukan Komparasi

Gambar 3.1 merupakan tahapan yang harus dilalui untuk melakukan kegiatan komparasi. Harapannya dengan adanya skema tahapan pada Gambar 3.1 akan membantu kegiatan penelitian menjadi lebih terarah dan berkesinambungan antara fase satu dan lainnya sehingga dapat mengurangi kesalahan yang mungkin terjadi dari aktivitas penelitian yang dilakukan.



Gambar 3.1 Tahap Komparasi.

3.2 Penentuan *Template* Dokumen SRS

Template dokumen SRS yang akan digunakan pada penelitian ini akan mengacu pada *template* yang diberikan oleh masing-masing standar. Namun, beberapa bagian dari *template* tersebut perlu dilakukan penyesuaian kembali agar tetap relevan dengan studi kasus yang diambil. Format dokumen SRS yang berasal dari dokumen MIL-STD-498 dapat dilihat pada Gambar 3.2 dan format IEEE 830-1998 dapat dilihat pada Gambar 3.3.

Khusus untuk IEEE, *best practice* ini menyediakan 8 *template* tambahan khusus untuk menyusun bab *specific requirements*. Pada penelitian ini dipilih *template* yang menyusun bab *specific requirements* berdasarkan fitur karena adanya sejumlah pertimbangan. Untuk mengetahui pertimbangan yang melatarbelakangi pemilihan tersebut dapat dilihat pada subbab 4.1.3 poin nomor 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 3.2 *Template* dokumen SRS MIL-STD-498

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

Gambar 3.3 *Template* dokumen SRS IEEE 830-1998

3.3 Pembuatan Dokumen SRS

Setelah ditentukan *template* dari masing-masing dokumen SRS, kemudian masuk pada fase selanjutnya yaitu pembuatan dokumen SRS. Selama proses pembuatan dokumen, penulis akan terus berkonsultasi dengan dosen ahli Rekayasa Perangkat Lunak di Universitas Islam Indonesia untuk mendapatkan masukan serta arahan pada proses pembuatan kedua dokumen ini. Hasil analisis kebutuhan yang dimuat pada kedua dokumen ini harus benar dan sesuai dengan kebutuhan perangkat lunak yang akan dikembangkan. Apabila terjadi perubahan kebutuhan dari sisi dokumen SRSnya maka perubahan tersebut juga akan diikuti dari sisi perangkat lunaknya. Setelah kedua dokumen SRS ini selesai dibuat kemudian akan diuji di fase selanjutnya yaitu pengujian dokumen.

3.4 Pengujian Dokumen SRS

Tujuan dilakukannya pengujian terhadap dokumen SRS ini ialah untuk mengetahui kualitas dari dokumen SRS dan menemukan berbagai kesalahan pada dokumen SRS yang telah dihasilkan. Adapun informasi yang ingin digali pada tahap pengujian ini, yaitu:

- a. Mencari tahu standar yang biasa digunakan oleh pihak ahli

Informasi ini penting untuk diketahui karena standar yang biasa digunakan pihak ahli dapat menjadi bahan pertimbangan dalam menentukan rekomendasi standar yang tepat untuk pengembang pemula. Adapun parameter yang digunakan dan pertanyaan pada kuisisioner yang mendukung parameter tersebut dapat dilihat pada Tabel 3.1 (detail pertanyaan dapat dilihat pada lampiran).

- b. Mencari tahu kualitas dari dokumen SRS yang dihasilkan

Kriteria kualitas dokumen SRS mengacu pada tesis yang ditulis oleh (Ali, 2006). Dari 21 karakteristik yang tersedia, dipilih 6 karakteristik sebagai parameter pengujian kualitas pada kedua dokumen SRS ini. Selain Ali, terdapat parameter lain yang juga digunakan untuk mendukung tujuan dilakukannya penelitian ini. Parameter tersebut adalah ringkas dan lengkap. Parameter ini digunakan untuk memberi rekomendasi kepada pengembang pemula mengenai standar yang dapat menyajikan dokumen SRS dengan kualitas yang ringkas namun tetap lengkap bagi mereka. Untuk mengetahui berbagai parameter kualitas tersebut dapat dilihat pada Tabel 3.2.

- c. Mencari tahu berbagai kesalahan yang muncul pada dokumen SRS

Informasi ini penting untuk diketahui mengingat dokumen SRS yang dihasilkan masih mungkin memiliki berbagai kesalahan. Oleh sebab itu kesalahan yang berhasil ditemukan pihak ahli harapannya dapat segera diperbaiki sehingga dokumen SRS yang dihasilkan dapat memenuhi parameter kualitas yang ada pada Tabel 3.2. Untuk mengetahui parameter yang digunakan dan pertanyaan pada kuisisioner yang mendukung parameter tersebut dapat dilihat pada Tabel 3.3 (detail pertanyaan dapat dilihat pada lampiran).

Untuk menggali ketiga informasi di atas maka pengujian akan dilakukan menggunakan kuisisioner. Dalam menjawab kuisisioner tersebut pihak penguji harus melakukan ulasan teknis (*technical review*) terhadap kedua dokumen SRS yang telah dihasilkan. Pihak yang ditunjuk untuk menguji dokumen SRS ini adalah dosen ahli Rekayasa Perangkat Lunak di Universitas Islam Indonesia.

Tabel 3.1 Parameter Untuk Mengetahui Standar yang Biasa Digunakan Oleh Pihak Ahli

No.	Parameter	Alasan Parameter Dipilih	Nomor Pertanyaan Pada Kuisisioner
1.	Standar yang umum digunakan	Untuk mengetahui standar yang biasanya digunakan pihak ahli dalam membuat dokumen SRS.	1, 2, 3

Tabel 3.2 Parameter Untuk Mengetahui Kualitas Dokumen SRS yang Dihasilkan

No.	Parameter	Alasan Parameter Dipilih	Nomor Pertanyaan Pada Kuisisioner
2.	Kebenaran	Untuk memastikan dokumen SRS yang dihasilkan telah benar dan sesuai dengan standar yg ada.	4, 11
3.	Kecukupan	Untuk memastikan dokumen SRS sudah cukup memenuhi standar penulisan dokumen SRS.	10
4.	Ambiguitas	Untuk memastikan bahwa setiap kebutuhan yang dinyatakan di kedua dokumen sudah jelas, mudah dipahami, dan tidak ambigu.	5, 6
5.	Bisa dimengerti	Untuk mengetahui standar yang lebih mudah dipahami oleh penguji.	15
6.	Keringkasan	Untuk memastikan kebutuhan yang dinyatakan pada kedua dokumen sudah dijelaskan secara singkat dan jelas.	7
7.	Kelengkapan	Untuk memastikan semua kebutuhan terkait perangkat lunak sudah dimuat di kedua dokumen SRS dan tidak ada kebutuhan yang terlewat.	8
8.	Keringkasan dan kelengkapan	Untuk mengetahui standar yang dapat memenuhi kualitas ringkas dan lengkap.	9

Tabel 3.3 Parameter Untuk Mengetahui Kesalahan Pada Dokumen SRS

No.	Parameter	Alasan Parameter Dipilih	Nomor Pertanyaan Pada Kuisisioner
9.	Kesalahan dan kekurangan pada dokumen	Untuk menggali informasi terkait berbagai kesalahan dan kekurangan yang ditemui pihak penguji pada dokumen SRS.	12, 13, 14

Perlu diketahui bahwa proses pengujian dokumen pada tahap ini tidak termasuk ke dalam pengujian *usability*. Pengujian *usability* baru akan dilakukan setelah dokumen SRS diuji dan dipastikan kebenaran dan kelengkapannya menurut para ahli.

3.5 Mengevaluasi Hasil Komparasi

Tahap akhir dari alur penelitian ini adalah mengevaluasi hasil komparasi. Fase ini berfungsi untuk memastikan kebenaran dan menilai tingkat kesesuaian hasil komparasi dengan mencocokkannya pada berbagai data yang telah diperoleh, seperti data hasil pengujian dokumen, pengujian *usability*, perangkat lunak dan sebagainya.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembuatan Dokumen SRS

Subbab ini membahas hal-hal terkait dokumen SRS yang telah dihasilkan, seperti hasil perbandingan antara kedua panduan, kelebihan kekurangan dari masing-masing panduan, kendala yang ditemui selama proses pembuatan dokumen SRS, perubahan yang terjadi disetiap draft dokumen SRS, dan hasil evaluasi pada dokumen SRS.

4.1.1 Hasil Dokumen SRS

Pembuatan dokumen SRS dilakukan oleh analis, dan diketahui bahwa format MIL-STD-498 menghasilkan dokumen SRS yang jauh lebih ringkas daripada format IEEE. Dengan studi kasus yang sama MIL-STD-498 hanya menghasilkan dokumen sebanyak 33 halaman, sedangkan IEEE 50 halaman. Ditambah lagi jumlah bab yang dimiliki oleh MIL-STD-498 jauh lebih banyak ketimbang IEEE. Untuk mengetahui hasil komparasi dari kedua panduan dan dokumen SRS tersebut dapat dilihat pada Tabel 4.1. Untuk mengetahui bagaimana struktur dokumen SRS dari format MIL-STD-498 dapat dilihat pada Gambar 4.1 hingga Gambar 4.2 dan Gambar 4.3 sampai Gambar 4.5 menampilkan stuktur dokumen SRS dari format IEEE.

Tabel 4.1 Hasil Komparasi Berdasarkan Panduan dan Dokumen SRS

No.	MIL-STD-498	IEEE 830-1998
1.	Dokumen SRS terbagi ke dalam 6 bab.	Dokumen SRS terbagi ke dalam 4 bab.
2.	Semua fungsi yang harus dimiliki perangkat lunak dijelaskan pada subbab “Kebutuhan Kemampuan Perangkat Lunak”	Semua fungsi yang harus dimiliki perangkat lunak dijelaskan pada subbab “Fungsionalitas”
3.	Fungsi-fungsi pada perangkat lunak dikelompokkan ke dalam modul-modul.	Fungsi-fungsi pada perangkat lunak disusun berdasarkan fungsionalitasnya.
4.	Berdasarkan dokumen panduan versi perangkat lunak yang dikembangkan harus disertakan.	Berdasarkan dokumen panduan tidak terdapat aturan untuk menyertakan versi perangkat lunak yang dikembangkan.
5.	Menghasilkan dokumen SRS sebanyak 33 halaman.	Menghasilkan dokumen SRS sebanyak 50 halaman.
6.	Tidak diharuskan memuat skenario masukan/keluaran di setiap fungsi yang ada.	Harus memuat skenario masukan/keluaran dari setiap fungsi yang ada.
7.	Walaupun tidak memiliki kebutuhan <i>state</i>	Tidak ada istilah <i>state</i> pada IEEE. Kebutuhan

	dan mode pada sistem, namun subbab ini tetap dimuat pada dokumen SRS.	mode juga diatur pada <i>template</i> yang berbeda dan kita bebas untuk menggunakannya atau tidak.
8.	Aturan untuk mengidentifikasi siapa saja pengguna sistem, terdapat pada subbab 1.2. Sedangkan penjelasan mengenai karakteristik pengguna tersebut ada di subbab berbeda yaitu 3.13.	Tidak ada aturan untuk mengidentifikasi siapa saja pengguna sistem. <i>Best practice</i> ini hanya mengharuskan kita menjelaskan karakteristik umum dari pengguna. Namun, informasi terkait siapa saja target pengguna dari sistem ini tetap dimuat dan dapat dilihat pada subbab 2.3.
9.	Tidak ada aturan yang mengharuskan untuk mengidentifikasi target pembaca dari dokumen ini. Namun informasi tersebut tetap ditambahkan pada subbab 1.3.	Terdapat aturan untuk mengidentifikasi siapa saja target pembaca dari dokumen ini. Informasi tersebut dapat dilihat pada subbab 1.1.
10.	Daftar Akronim dan Glosarium terdapat di bab 6 (subbab 6.1 dan 6.2).	Daftar definisi dan singkatan terdapat di bab 1 (subbab 1.3).
11.	Tidak memuat informasi tentang kebutuhan performa.	Memuat informasi tentang kebutuhan performa.
12.	Bab 1 memiliki 3 subjek bahasan	Bab 1 memiliki 5 subjek bahasan
13.	Bab 2 memiliki 1 subjek bahasan	Bab 2 memiliki 5 subjek bahasan
14.	Tidak menyediakan pilihan <i>template</i> lain untuk mengatur bab 3.	Menyediakan 8 <i>template</i> tambahan khusus untuk mengatur bab 3.
15.	Memiliki kebutuhan antarmuka eksternal dan internal.	Memiliki kebutuhan antarmuka eksternal saja.
16.	Menggunakan istilah “Kebutuhan” pada subbab kebutuhan perangkat keras komputer, kebutuhan perangkat lunak komputer, kebutuhan komunikasi komputer, dan informasi tersebut masuk ke dalam subbab “Kebutuhan Sumber Daya Komputer”.	Menggunakan istilah “Antarmuka” pada subbab antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi, dan informasi tersebut masuk ke dalam subbab Kebutuhan Antarmuka Eksternal.
17.	Memiliki kebutuhan keamanan maupun kerahasiaan. Misalnya pada subbab 1.3 “Ulasan Dokumen”, dan subbab 3.8 “Persyaratan keamanan dan kerahasiaan”	Hanya memiliki kebutuhan keamanan.

Dari tabel 4.1 terlihat sejumlah perbedaan dari kedua dokumen SRS tersebut. Misalnya, IEEE tidak mempunyai aturan untuk mengidentifikasi target pengguna dari perangkat lunak sebagaimana MIL-STD-498. Sebaliknya, MIL-STD-498 tidak memiliki aturan untuk mengidentifikasi target pembaca dari dokumen SRS seperti halnya IEEE. Namun informasi tersebut tetap ditambahkan baik pada MIL-STD-498 maupun IEEE karena sebisa mungkin segala perbedaan yang muncul dari kedua dokumen ini dibuat seragam agar memudahkan pihak analis dalam membuat dokumen SRS dan melakukan proses perbandingan antara dokumen satu dengan yang lainnya.

Akan tetapi tidak semua hal dapat diseragamkan. Dari Tabel 4.1 dapat dilihat beberapa informasi yang masih berbeda. Contohnya, tidak terdapat subbab *state* dan mode di IEEE

walaupun pada MIL-STD-498 subbab tersebut harus ada. Atau pada IEEE pembahasan mengenai antarmuka perangkat keras, perangkat lunak, maupun komunikasi terdapat pada subbab kebutuhan antarmuka eksternal, sedangkan pada MIL-STD-498 pembahasan tersebut ada di subbab kebutuhan sumber daya komputer. Perbedaan tersebut sulit untuk diseragamkan karena memang struktur maupun aturan di kedua *template* tersebut tidak sama. Penyeragaman hanya dilakukan jika informasi yang ditambah atau diganti tidak mengubah struktur pada dokumen baik dari MIL-STD-498 maupun IEEE. Alhasil beberapa perbedaan tetap dibiarkan seperti yang dapat dilihat pada nomor 2, 3, 6, 7, 10, 11, 12, 13, 15, 16, 17 di Tabel 4.1

DAFTAR ISI	
DAFTAR PERUBAHAN	iii
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	ix
1. Cakupan	1
1.1. Identifikasi	1
1.2. Gambaran Sistem.....	1
1.3. Gambaran Dokumen.....	1
2. Dokumen Referensi	2
3. Kebutuhan (<i>Requirement</i>).....	2
3.1. Kebutuhan <i>State</i> dan <i>Mode</i>	2
3.2. Kebutuhan Kemampuan Perangkat Lunak	2
3.2.1 Modul Layanan Permohonan.....	2
3.2.2. Modul Pengelolaan Data Induk	6
3.3. Kebutuhan Antarmuka Eksternal Perangkat Lunak	8
3.4. Kebutuhan Antarmuka Internal Perangkat Lunak	10
3.5. Kebutuhan Data Internal Perangkat Lunak.....	10
3.6. Kebutuhan Adaptasi.....	11
3.7. Kebutuhan Keamanan.....	11
3.8. Persyaratan Keamanan dan privasi	11
3.9. Kebutuhan lingkungan Perangkat Lunak.....	12
3.10. Kebutuhan Sumber Daya Komputer.....	12
3.10.1. Kebutuhan Perangkat Keras Komputer.....	12
3.10.2. Kebutuhan Penggunaan Sumber Daya Perangkat Keras Komputer	12
3.10.3. Kebutuhan Perangkat Lunak Komputer.....	12
3.10.4. Kebutuhan Komunikasi Komputer	13
3.11. Faktor Kualitas Perangkat Lunak	13
3.11.1. Keandalan.....	13
3.11.2. Perawatan	13
3.11.3. Ketersediaan.....	14
3.11.4. Fleksibilitas	14
3.11.5. <i>Portability</i>	14

Dokumen Ini Menggunakan Format MIL-STD-498

Gambar 4.1 Daftar Isi Dokumen SRS Format MIL-STD-498 Bagian 1

3.12. Batasan Implementasi dan Desain	15
3.13. Kebutuhan Terkait Pengguna	15
3.14. Kebutuhan Terkait Pelatihan	16
3.15. Kebutuhan Terkait Logistik	16
3.16. Kebutuhan Lain	16
3.17. Kebutuhan Pengemasan	17
3.18. Kebutuhan Kritis dan Diutamakan	17
4. Ketentuan Kualifikasi	17
5. Kebutuhan Ketelusuran	18
5.1. Pelacakan Kebutuhan	18
5.2. Pelacakan Kebutuhan Spesifik	20
6. Catatan	22
6.1. Daftar Akronim	22
6.2. Glosarium	22

Dokumen Ini Menggunakan Format MIL-STD-498

Gambar 4.2 Daftar Isi Dokumen SRS Format MIL-STD-498 Bagian 2

Daftar Isi	
Daftar Perubahan	i
Daftar Isi	iv
Daftar Tabel	vii
Daftar Gambar	viii
1. Pendahuluan	1
1.1. Tujuan	1
1.2. Ruang Lingkup Perangkat Lunak	1
1.3. Definisi, Singkatan	1
1.4. Referensi	3
1.5. Ikhtisar	3
2. Deskripsi Keseluruhan	5
2.1. Perspektif Produk	5
2.2. Fungsionalitas Produk	5
2.3. Karakteristik Pengguna	6
2.4. Batasan	7
2.5. Asumsi dan Ketergantungan	7
3. Kebutuhan Spesifik	9
3.1. Kebutuhan Antarmuka Eksternal	9
3.1.1. Antarmuka Pengguna	9
3.1.2. Antarmuka Perangkat Keras	11
3.1.3. Antarmuka Perangkat Lunak	11
3.1.4. Antarmuka Komunikasi	11
3.2. Fungsionalitas	11
3.2.1. Mengajukan Permohonan Peminjaman Perlengkapan Acara	12
3.2.1.1. Stimulus / Respon	12
3.2.1.2. Kebutuhan Fungsionalitas	13
3.2.2. Mengajukan Permohonan Peminjaman Sarana Pertemuan	13
3.2.2.1. Stimulus / Respon	13
3.2.2.2. Kebutuhan Fungsionalitas	14
3.2.3. Mengajukan Permohonan Konsumsi	14
3.2.3.1. Stimulus / Respon	15
3.2.3.2. Kebutuhan Fungsionalitas	16
3.2.4. Mengajukan Permohonan Akomodasi	16
3.2.4.1. Stimulus / Respon	16
3.2.4.2. Kebutuhan Fungsionalitas	17
Dokumen ini menggunakan format IEEE 830 - 1998	

Gambar 4.3 Daftar Isi Dokumen SRS Format IEEE 830-1998 Bagian 1

3.2.5. Mengajukan Permohonan Penyediaan Barang Habis Pakai	17
3.2.5.1. Stimulus / Respon.....	17
3.2.5.2. Kebutuhan Fungsionalitas	18
3.2.6. Mengajukan Permohonan Pemeliharaan Bangunan	19
3.2.6.1. Stimulus / Respon.....	19
3.2.6.2. Kebutuhan Fungsionalitas	20
3.2.7. Melakukan otorisasi Permohonan Peminjaman Perlengkapan Acara	20
3.2.7.1. Stimulan / Respon	20
3.2.7.2. Kebutuhan Fungsionalitas	21
3.2.8. Melakukan Otorisasi Permohonan Peminjaman Sarana Pertemuan	21
3.2.8.1. Stimulus / Respon.....	21
3.2.8.2. Kebutuhan Fungsionalitas	22
3.2.9. Melakukan Otorisasi Permohonan Konsumsi	22
3.2.9.1. Stimulus / Respon.....	23
3.2.9.2. Kebutuhan Fungsionalitas	23
3.2.10. Melakukan Otorisasi Permohonan Akomodasi	24
3.2.10.1. Stimulus / Respon.....	24
3.2.10.2. Kebutuhan Fungsionalitas	25
3.2.11. Melakukan Otorisasi Permohonan Penyediaan Barang Habis Pakai	25
3.2.11.1. Stimulus / Respon.....	25
3.2.11.2. Kebutuhan Fungsionalitas	26
3.2.12. Melakukan Otorisasi Permohonan Pemeliharaan Bangunan	26
3.2.12.1. Stimulan / Respon	26
3.2.12.2. Kebutuhan Fungsionalitas	27
3.2.13. Melihat Status Permohonan	27
3.2.13.2. Kebutuhan Fungsionalitas	28
3.2.14. Melihat Riwayat Pengajuan Permohonan	28
3.2.14.1. Stimulan / Respon	28
3.2.14.2. Kebutuhan Fungsionalitas	29
3.2.15. Melihat Riwayat Otorisasi Permohonan	29
3.2.15.1. Stimulus / Respon.....	29
3.2.15.2. Kebutuhan Fungsionalitas	29
3.2.16. Mengubah Hak Akses Pengguna	29
3.2.16.1. Stimulus / Respon.....	30
3.2.16.2. Kebutuhan Fungsionalitas	30
3.2.17. Mengelola Item Layanan	30
3.2.17.1. Stimulus / Respon.....	30

Dokumen ini menggunakan format IEEE 830 - 1998

Gambar 4.4 Daftar Isi Dokumen SRS Format IEEE 830-1998 bagian 2

3.2.17.2. Kebutuhan Fungsionalitas	31
3.2.18. Mengelola Data Departemen	31
3.2.18.1. Stimulus / Respon	31
3.2.18.2. Kebutuhan Fungsionalitas	32
3.2.19. Mengelola Data Aktor.....	32
3.2.19.1. Stimulus / Respon.....	32
3.2.19.2. Kebutuhan Fungsionalitas	33
3.2.20. Mengelola Alur Otorisasi.....	33
3.2.20.1. Stimulus / Respon.....	34
3.2.20.2. Kebutuhan Fungsionalitas.....	34
3.2.21. Mengunduh Permohonan	35
3.2.21.1. Stimulus / Respon.....	35
3.2.21.2. Kebutuhan Fungsionalitas	35
3.3 Kebutuhan Performa.....	35
3.4. Standar Kepatuhan.....	36
3.5. Atribut Sistem Perangkat Lunak.....	36
3.5.1. Keandalan	36
3.5.2. Ketersediaan.....	36
3.5.3. Keamanan	37
3.5.4. Perawatan	37
3.5.5. Portability	38
3.6. Kebutuhan Basisdata	38
4. Informasi Dukungan.....	39
4.1. Pelacakan Kebutuhan.....	39
4.2. Pelacakan Kebutuhan Spesifik	40

Dokumen ini menggunakan format IEEE 830 - 1998

Gambar 4.5 Daftar Isi Dokumen SRS Format IEEE 830-1998 Bagian 3

4.1.2 Kelebihan dan Kekurangan

Terdapat sejumlah kelebihan maupun kekurangan yang ditemui pada standar MIL-STD-498 dan IEEE 830-1998. Berikut uraiannya:

a. MIL-STD-498

Kelebihan yang dimiliki oleh standar ini ialah:

1. Gratis

Standar yang disusun oleh Departemen Pertahanan Amerika Serikat ini bersifat *public domain*. Artinya siapa saja bisa mendapatkan serta menggunakannya secara gratis.

2. Lengkap

Seluruh panduan dan *template* dokumen yang dibutuhkan selama proses pengembangan perangkat lunak mulai dari SDP, SRS, IRS, SDD, STP, dan lainnya telah tersedia di dalam satu standar. Sehingga kita tidak perlu lagi mencari dokumen panduan atau *template* yang berasal dari standar lain.

3. Disertai penjelasan

Tidak hanya terdiri dari *template* saja (yang berbentuk daftar isi), standar ini juga dilengkapi deskripsi disetiap subjek bahasannya sehingga memudahkan analisis pada proses pembuatan dokumen SRS.

4. Dokumen yang dihasilkan lebih ringkas

Karena MIL-STD-498 tidak perlu memuat skenario masukan/keluaran disetiap fungsionalitas perangkat lunak seperti halnya IEEE, ditambah lagi subjek bahasan pada bab 1 dan bab 2 jauh lebih sederhana, maka dokumen SRS yang dihasilkan pun menjadi lebih ringkas daripada dokumen SRS format IEEE.

Kekurangan yang dimiliki oleh standar ini ialah:

1. Penjelasan pada subjek bahasan sulit dipahami

Walau standar ini memberikan penjelasan disetiap subjek bahasannya, namun penjelasan tersebut seringkali sulit dipahami dan cenderung membingungkan. Terutama pada istilah-istilah teknis yang digunakan. Kurangnya penjelasan eksplisit di beberapa subjek bahasan merupakan kelemahan pada standar ini sehingga cukup menyulitkan pihak pembuat dokumen SRS khususnya pengembang pemula.

2. Jarang digunakan

Standar ini jarang digunakan kecuali untuk keperluan militer atau sistem dengan tingkat kerumitan dan bahaya yang tinggi. Alhasil referensi dokumen SRS yang diperoleh pun tidak banyak membantu proses pembuatan dokumen karena pembahasan yang dimuat pada contoh tersebut kurang familier dan sulit untuk dipahami.

b. IEEE 830-1998

Kelebihan yang dimiliki oleh *best practice* ini ialah:

1. Banyak digunakan

IEEE 830-1998 merupakan panduan yang paling banyak digunakan dalam membuat dokumen SRS. Khususnya untuk kebutuhan pendidikan (kuliah). Sehingga mencari referensi terkait *best practice* ini seperti deskripsi dari setiap subjek bahasan maupun contoh dokumennya akan sangat mudah ditemukan, baik yang menggunakan Bahasa Inggris maupun Bahasa Indonesia.

2. Dilengkapi penjelasan

Sama seperti MIL-STD-498, standar IEEE juga dilengkapi dengan deskripsi disetiap subjek bahasannya sehingga membantu pihak analis pada proses pembuatan dokumen SRS.

3. Lebih mudah dipahami

Tidak seperti MIL-STD-498, penjelasan atau deskripsi yang diberikan oleh *best practice* ini lebih mudah untuk dipahami.

4. Menyediakan informasi tentang kualitas dokumen SRS yang baik

IEEE tidak hanya menyediakan *template* beserta penjelasan disetiap subjek bahasannya. Namun juga memberikan informasi mengenai hal-hal yang harus dipertimbangkan saat menulis dokumen SRS agar dokumen yang dihasilkan dapat memiliki kualitas yang baik.

5. Menyediakan banyak *template* untuk bab kebutuhan spesifik

Terkadang perangkat lunak memiliki karakteristik yang berlainan. Contohnya, ada perangkat lunak yang memiliki banyak aktor dan setiap aktor mempunyai fungsi yang berbeda. Ada juga perangkat lunak yang memiliki lebih dari satu mode operasi. Untuk memenuhi karakteristik yang berbeda-beda tersebut IEEE menyediakan 8 *template* tambahan khusus untuk menyusun bab Kebutuhan Spesifik agar dokumen yang dihasilkan menjadi lebih rinci, sesuai dengan karakteristik perangkat lunak yang dikembangkan dan mudah dipahami pembaca.

6. Struktur *template* lebih rapi

Walau subjek bahasan bab 1 dan bab 2 IEEE lebih banyak ketimbang MIL-STD-498 namun strukturnya lebih detail dan rapi. Pada bab 1 IEEE berbicara hal-hal terkait dokumen sedangkan bab 2 membahas sistem dan latar belakang informasi dari

sistem tersebut. Informasi antara dokumen dan sistem tidak dicampur sehingga memudahkan proses pencarian informasi dari sisi pembaca.

Kekurangan yang dimiliki *best practice* ini ialah:

1. Berbayar

Tidak seperti MIL-STD-498 yang bisa diperoleh dengan gratis, untuk dapat menggunakan *best practice* ini kita harus membelinya terlebih dahulu.

2. Dokumen SRS lebih banyak

Format ini menghasilkan dokumen SRS yang jauh lebih banyak daripada MIL-STD-498. Hal itu disebabkan salah satunya karena IEEE harus memuat skenario tentang masukan/keluaran dari setiap fungsionalitas yang ada. Kemudian pembahasan pada bab 1 dan bab 2 format ini juga jauh lebih banyak daripada MIL-STD-498 sehingga mempengaruhi keringkasan pada dokumen SRS yang dihasilkan.

4.1.3 Kendala Pembuatan Dokumen SRS

Ada beberapa kendala yang dihadapi selama proses pembuatan dokumen SRS, baik dari format MIL-STD-498 maupun IEEE 830-1998. Kendala tersebut adalah:

- a. Kurangnya Penjelasan Eksplisit

Kurangnya penjelasan eksplisit yang diberikan MIL-STD-498 maupun IEEE 830-1998 di beberapa istilah teknis maupun subjek bahasan menjadi kendala utama yang menyebabkan proses pembuatan dokumen SRS membutuhkan waktu yang lebih lama. Hal itu dikarenakan pihak analis tidak mendapatkan gambaran yang jelas mengenai apa yang harus disampaikan pada dokumen SRS ini.

Contohnya istilah CSCI pada standar MIL-STD-498. Standar itu sebenarnya telah menjelaskan makna dari istilah CSCI, namun penjelasan yang disediakan tidak dapat memberi gambaran konkret mengenai maksud dari istilah tersebut, apakah CSCI diartikan sebagai suatu perangkat lunak, atau suatu kebutuhan, ataukah yang lainnya.

Contoh lainnya penggunaan istilah *state* dan *mode* masih pada standar yang sama. MIL-STD-498 tidak memberikan definisi pasti mengenai apa yang dimaksud *state* dan *mode*. MIL-STD-498 hanya mengatakan bahwa pemisahan kedua istilah tersebut dilakukan secara *arbitrari* (sewenang-wenang) dan suatu CSCI dapat dideskripsikan ke dalam *state* saja, atau *mode* saja, atau *state* di dalam *mode*, atau *mode* di dalam *state* tanpa tahu apa maksud dan perbedaan dari ke dua istilah tersebut.

Ambiguitas dan kebingungan yang muncul ketika membaca penjelasan kedua panduan tersebut dapat menyebabkan kesalahan dalam menafsirkan makna dari istilah teknis yang digunakan maupun subjek bahasan pada dokumen panduan. Akibatnya dokumen SRS yang dihasilkan pun menjadi tidak benar, tidak lengkap, atau memiliki kualitas yang rendah. Kendala ini khususnya akan dirasakan apabila pihak yang membuat dokumen SRS merupakan pengembang pemula yang belum memiliki banyak pengalaman dalam membuat perangkat lunak maupun dokumen SRS.

Guna mengatasi masalah tersebut, pendekatan yang dilakukan yaitu berdiskusi dengan pihak pembimbing mengenai maksud dari istilah teknis maupun subjek bahasan yang tidak dipahami. Selain itu beberapa istilah teknis yang membingungkan juga disepakati untuk diubah baik pada MIL-STD-498 maupun IEEE agar proses pembuatan dokumen SRS menjadi lebih mudah. Pendekatan ini akan dibahas lebih lanjut pada poin no 2.

b. Banyaknya Istilah Teknis yang Digunakan

Jika kita mengambil contoh pada format IEEE 830-1998 terlihat cukup banyak istilah teknis yang digunakan seperti fungsi, fitur, kebutuhan fungsional, *associate functional requirements*, dan lainnya. Untuk sistem dengan skala besar terkadang memang perlu memisahkan istilah-istilah tersebut ke dalam kelompok-kelompok berbeda agar dokumen yang dihasilkan menjadi lebih rinci. Namun untuk sistem berskala kecil dan level kerumitan yang lebih rendah, hal itu seringkali tidak dibutuhkan sebab banyaknya istilah yang digunakan malah cenderung menyulitkan dan membingungkan pihak pengembang pada proses pembuatan dokumen SRS. Oleh sebab itu, untuk mengurangi tingkat kerumitan saat membuat dokumen SRS serta menyederhanakan isi dokumennya maka penggunaan sejumlah istilah teknis baik pada MIL-STD-498 maupun IEEE 830-1998 disepakati untuk diminimalisir. Caranya dengan mengganti beberapa istilah bawaan pada dokumen panduan dengan istilah baru yang memiliki makna sepadan namun lebih familier atau mudah dipahami terutama oleh pengembang pemula. Agar lebih jelas, Tabel 4.2 menampilkan daftar istilah-istilah teknis yang telah diubah baik pada format MIL-STD-498 maupun IEEE.

Tabel 4.2 Daftar Istilah Teknis yang Disesuaikan

No.	Format	Istilah Lama	Istilah Baru
1.	MIL-STD-498	CSCI	Perangkat Lunak
		<i>Acquirer</i> (pengada)	Pelanggan
		<i>Developer</i> (pengembang)	Pengembang

2.	IEEE 830-1998	Kekangan	Batasan
		<i>Customer</i> (pelanggan)	Pelanggan
		<i>Supplier</i> (pemasok)	Pengembang
		Fungsi Produk	Fungsionalitas Produk
		Fitur Sistem	Fungsionalitas
		Tujuan Fitur	Tujuan Fungsionalitas
		Kebutuhan Fungsional	Kebutuhan Fungsionalitas

c. Perlunya menentukan kebutuhan performa

Ketika menentukan kebutuhan performa di dokumen SRS versi IEEE, kendala yang dihadapi adalah bagaimana cara menetapkan kebutuhan performa dari suatu perangkat lunak. Adakah aturan, panduan, maupun standar untuk menentukan kebutuhan tersebut ataukah cukup dengan menanyakan kebutuhan performa yang diinginkan langsung kepada pelanggan. Pendekatan yang digunakan untuk menyelesaikan masalah ini yaitu dengan mengacu pada hasil penelitian Robert B. Miller yang dikutip Jakob Nielsen mengenai batas waktu respon sistem yang masih ditoleransi oleh pengguna. Dari penelitian itu kemudian mayoritas kebutuhan performa pada perangkat lunak ini ditetapkan sebesar 1 detik, sebab menurut penelitian tersebut diangka itu kendati pengguna merasakan adanya respon yang terlambat dari sistem namun mereka masih menganggap hal itu bukan sebagai suatu masalah. Akan tetapi ada juga kebutuhan yang ditetapkan dibawah 10 detik karena pertimbangan tingkat kerumitan dari fungsionalitas tersebut.

d. Perlunya Menentukan *template* untuk bab kebutuhan Spesifik

Kendala sewaktu menentukan *template* untuk menyusun bab kebutuhan spesifik di dokumen SRS versi IEEE adalah, dari kedelapan *template* yang tersedia manakah *template* yang tepat untuk digunakan dan apa pertimbangan yang mendasari dipilihnya *template* tersebut. Untuk mengatasi kendala itu, analisis berpatokan pada deskripsi yang ada di masing-masing *template* serta mencari contoh dokumen SRS lain yang mengaplikasikan *template-template* tersebut. Hal ini dilakukan untuk memberikan pemahaman yang lebih baik tentang cara menyusun bab kebutuhan spesifik di setiap *template* dan menjadikan contoh-contoh yang diperoleh sebagai pertimbangan saat memilih *template* yang akan digunakan.

Setelah contoh diperoleh, kemudian seluruh *template* yang ada diseleksi untuk mengetahui *template* mana yang dapat digunakan. Caranya dengan mempelajari deskripsi di setiap *template* dan contoh dokumen SRS yang telah didapat, kemudian

mencocokkannya dengan karakteristik perangkat lunak yang akan dikembangkan. Dari delapan *template* yang ada, dipilihlah dua *template* yang paling mungkin untuk digunakan yaitu, *user class* dan fitur. Alasannya, *template user class* dipilih karena perangkat lunak yang dijelaskan pada dokumen ini mempunyai aktor yang beragam dengan peran yang berbeda-beda. Sedangkan *template* fitur dipilih karena perangkat lunak ini menyediakan berbagai fitur berbeda guna memenuhi kebutuhan penggunaanya. Namun, dari kedua *template* tersebut akhirnya pilihan jatuh kepada *template* fitur. karena, *template user class* hanya dapat digunakan jika setiap aktor memiliki fungsi yang berbeda-beda. Sedangkan di perangkat lunak ini beberapa aktor masih memiliki fungsi yang sama dengan aktor lainnya. Kemudian *template* fitur tidak memiliki syarat tertentu untuk dapat digunakan, cukup menjelaskan fitur apa yang dimiliki oleh perangkat lunak dan kebutuhan fungsional dari fitur-fitur tersebut. Selain itu struktur pada *template* ini juga sudah cukup untuk menjelaskan semua kebutuhan yang ada di perangkat lunak. Atas pertimbangan tersebut maka dipilihlah *template* fitur untuk menyusun bab kebutuhan spesifik di dokumen SRS ini. Akan tetapi untuk menyederhanakan isi pada dokumennya beberapa perubahan tetap perlu dilakukan, seperti menghilangkan subbab *associated functional requirement* dan meminimalisir penggunaan istilah-istilah teknis seperti yang telah disebutkan pada poin a dan b.

4.1.4 Rangkuman Perubahan Dokumen SRS

Terdapat 5 draft dokumen yang dihasilkan selama proses pembuatan dokumen SRS baik dari standar MIL-STD-498 dan IEEE. Keempat draft tersebut dihasilkan guna menyempurnakan draft dokumen versi sebelumnya. Untuk mengetahui apa saja perubahan yang terjadi di setiap draft pada kedua versi dokumen tersebut dapat dilihat rangkumannya pada Tabel 4.3 dan

Tabel 4.4.

Tabel 4.3 Rangkuman Perubahan Dokumen SRS format IEEE 830-1998

No.	Nomer Draft	Daftar Perubahan		Bab yang Diubah
1.	Draft I	a.	Konten fungsi produk (subbab 2.2) dan fitur sistem (subbab 3.2) terlihat tidak memiliki perbedaan, sehingga menimbulkan ambiguitas dan kesan pengulangan informasi ketika dokumen ini dibaca. Untuk itu bagian fungsi produk diubah ke dalam bentuk tabel sederhana dan diisi dengan fungsi apa saja yang dimiliki oleh perangkat lunak tanpa	2.2

			diberikan penjelasan.	
		b.	Penggunaan kata kekangan sebagai judul subbab 2.4 diubah menjadi batasan.	2.4
		c.	Menghapus <i>mockup</i> yang dimuat pada subbab kebutuhan antarmuka pengguna, dan mengubahnya ke dalam bentuk deskripsi (penjelasan/tulisan) mengenai syarat yang harus dipenuhi sistem terkait kebutuhan antarmuka pengguna tersebut. (beberapa <i>mockup</i> masih tetap dimuat pada dokumen ini dengan alasan memberikan gambaran kepada pembaca mengenai tampilan perangkat lunak yang akan dikembangkan)	3.1.1
		d.	Kebutuhan basisdata tidak lagi diisi dengan diagram ERD. Melainkan tabel sederhana yang isinya berupa data apa saja yang akan disimpan atau dibutuhkan oleh perangkat lunak.	3.6
		e.	Fitur <i>Login</i> , <i>Lupa Password</i> , dan <i>Ganti Password</i> tidak lagi dimasukkan ke subbab fitur sistem. Melainkan dimasukkan ke dalam subbab kebutuhan keamanan.	3.5.3
2.	Draft II	a.	Mengubah judul subbab fungsi produk menjadi fungsionalitas produk.	2.2
		b.	Mengubah judul subbab fitur sistem menjadi fungsionalitas.	3.2
		c.	Mengubah judul subbab tujuan fitur menjadi tujuan fungsionalitas.	3.2.n.1
		d.	Mengubah judul subbab kebutuhan fungsional menjadi kebutuhan fungsionalitas.	3.2.m.3
3.	Draft III	a.	Menambahkan tabel pelacakan kebutuhan	4.1
		b.	Menghapus kolom kode fungsionalitas beserta isinya pada tabel fungsionalitas produk	2.2
		c.	Menghapus kode fungsionalitas di semua judul subbab fungsionalitas	3.2.x
		d.	Menambahkan kalimat pengantar di beberapa subbab	1.3, 1.4, 2.3, 3.1, 3.2, 3.5.
4.	Draft IV	a.	Menambah metode konektivitas antara perangkat lunak dengan basisdata pusat PT. Pupuk Kaltim.	2.5, 3.1.3
		b.	Menambahkan penanganan kesalahan pada entitas luar.	3.5.4
		c.	Memperbaiki kebutuhan keamanan perangkat lunak	3.5.3
		d.	Memperbaiki inkonsistensi istilah “Pre-kondisi” dan “Post-kondisi” di tabel 3.1 – 3.21	3.2.1.x
		e.	Memperbaiki kebutuhan fungsionalitas terkait alur prosedur otorisasi	3.2.7.2 – 3.2.12.2
		f.	Memperbaiki SOP pada fungsi login dan menghapus fungsi ganti password, lupa password, dan metode server side validation dan client side validation	3.5.3
		g.	Menambahkan tabel Pelacakan Kebutuhan Spesifik	4.2
		h.	Memperbaiki kesalahan pengetikan	
5.	Draft V	a	Menambahkan nomor dokumen, kebutuhan keamanan, dan pernyataan distribusi.	Sampul
		b	Menambahkan informasi mengenai perwakilan pengembang yang merilis dokumen, dan perwakilan pelanggan yang menyetujui dokumen	Informasi Administrasi
		c	Menambahkan kebutuhan antarmuka perangkat lunak	3.3.1.
		d	Menambahkan kebutuhan antarmuka pengguna	3.3.2

	e	Menambahkan kebutuhan perangkat keras komputer	3.10.1
	f	Memperbaiki kebutuhan pada faktor kualitas kegunaan	3.11.7
	g	Memperbaiki Kebutuhan terkait pengguna	3.13
	h	Menambahkan detail kebutuhan terkait pelatihan	3.14
	i	Memperbaiki Ketantuan kualifikasi	4

Tabel 4.4 Rangkuman Perubahan Dokumen SRS format MIL-STD-498

No.	Nomer Draft	Daftar Perubahan		Bab yang Diubah
1.	Draft I	a.	Kebutuhan kemampuan CSCI dikelompokkan ke dalam modul-modul.	3.2
		b.	Menambahkan data yang dibutuhkan oleh perangkat lunak pada subbab kebutuhan data internal perangkat lunak.	3.5
		c.	Fitur <i>Login</i> , <i>Lupa Password</i> , dan <i>Ganti Password</i> dimasukkan ke dalam kebutuhan keamanan.	3.7
2.	Draft II	a.	Berdasarkan kesepakatan maka istilah CSCI diganti menjadi “perangkat lunak”, sehingga kata CSCI tidak lagi digunakan diseluruh dokumen ini.	Seluruh dokumen
		b.	Mengganti istilah kebutuhan perangkat lunak menjadi kebutuhan perangkat lunak komputer.	3.10.3
		c.	Menambahkan tabel pelacakan kebutuhan yang isinya mengacu pada bab 3.2.	5
		d.	Sampul pada dokumen SRS disamakan dengan sampul pada dokumen SRS versi IEEE.	Sampul
3.	Draft III	a.	Mengganti kalimat <i>software requirement specification</i> menjadi spesifikasi kebutuhan perangkat lunak pada halaman sampul.	Sampul
		b.	Menambahkan kolom nomor untuk tabel pelacakan kebutuhan	5
		c.	Menambahkan kalimat pengantar ke beberapa subbab	5, 6.1, 6.2
4.	Draft IV	a.	Menambah metode konektivitas antara perangkat lunak dengan basisdata pusat PT. Pupuk Kaltim.	3.3
		b.	Menambahkan penanganan kesalahan pada entitas luar.	3.11.2
		c.	Konten pada subbab kebutuhan keamanan dipindah ke subbab persyaratan keamanan dan privasi	3.8
		d.	Memperbaiki SOP pada fungsi login dan menghapus fungsi ganti password, lupa password, dan motide server side validation dan client side validation	3.7
		e.	Mengganti kebutuhan <i>portability</i>	3.11.5
		f.	Mengganti kebutuhan <i>reusability</i>	3.11.6
5.	Draft V	a.	Menambahkan definisi terkait perlengkapan acara, sarana pertemuan, konsumsi, akomodasi, barang habis pakai, dan pemeliharaan bangunan di Tabel 1.1.	1.3
		b.	Memperbaiki subbab batasan	2.4
		c.	Memperbaiki subbab asumsi dan kebutuhan	2.5
		d.	Menambahkan kebutuhan antarmuka perangkat keras	3.1.2
		e.	Memperbaiki kebutuhan di subbab kebutuhan performa	3.3

4.1.5 Evaluasi Dokumen

Dokumen SRS dan perangkat lunak yang menjadi objek penelitian ini telah dibuat sebelum kegiatan riset dilakukan. Akan tetapi dokumen SRS yang dihasilkan masih belum sesuai dengan standar yang ada. Oleh sebab itu dokumen SRS tersebut kemudian diperbaharui mengikuti standar MIL-STD-498 dan IEEE 830-1998. Proses pembaharuan itu mengakibatkan sejumlah perubahan dan perbaikan dari sisi dokumen SRS, sehingga perlu dilakukan evaluasi kembali terhadap perangkat lunaknya untuk memeriksa tingkat kesesuaiannya dengan dokumen SRS yang baru. Setiap perubahan kebutuhan pada dokumen SRS akan diikuti perubahan pada perangkat lunak. Namun perubahan yang dilakukan tidak melalui tahap desain komprehensif hingga menghasilkan dokumen SDD seperti pada umumnya karena fase itu ada diluar ruang lingkup penelitian ini. Sebagai gantinya hanya dibuat beberapa rancangan sederhana untuk membantu proses pembaharuan pada perangkat lunak seperti membuat diagram *usecase* dan prototipe antarmuka pengguna.

Hasil evaluasi menunjukan bahwa perangkat lunak telah memenuhi sebagian besar kebutuhan di dokumen SRS. Namun, masih ada beberapa kebutuhan lain yang belum bisa dipenuhi karena sejumlah alasan. Untuk mengetahui kebutuhan apa saja yang belum dapat dipenuhi oleh perangkat lunak dapat dilihat pada Tabel 4.5.

Tabel 4.5 Daftar Kebutuhan yang Gagal di Implementasikan

No.	Kebutuhan pada dokumen SRS	Implementasi pada sistem
1.	Admin dapat mengelola (tambah, ubah, hapus) seluruh data layanan tanpa perlu mengakses basisdata.	Kebutuhan tersebut tidak dapat dipenuhi karena terkendala desain basisdata dan antarmuka pengguna. Jika admin ingin menambah data layanan ia perlu membuat tabel baru pada basisdata dengan atribut yang dibutuhkan layanan tersebut, kemudian menambahkan nama layanan itu ke tabel <i>service</i> yang ada di basisdata. Setelah selesai kemudian perlu dilakukan modifikasi pada antarmukanya sehingga dapat memunculkan menu layanan yang baru ditambahkan tadi. Proses inilah yang menyebabkan kebutuhan ini tidak dapat dipenuhi oleh sistem.
2.	Admin dapat mengubah dan menghapus data aktor.	Beberapa data aktor memang dapat diubah dan dihapus kapan saja. Namun untuk data aktor Manager, KEPALA SEKSI, KEPALA BAGIAN, MANAGER UMUM, dan SEKRETARIS tidak dapat diubah atau dihapus secara mudah oleh Admin. Sebab data tersebut digunakan di dalam kode program (statis), sehingga Admin harus mengganti atau menghapus data yang diinginkan pada basisdata setelah itu mengubah kode pada

		program agar fungsionalitas pada sistem tetap berjalan.
3.	Sekretaris DPU yang berstatus sebagai karyawan organik dapat mengajukan permohonan layanan. Sedangkan Sekretaris DPU yang berstatus karyawan <i>outsourcing</i> tidak dapat mengajukan permohonan layanan.	Pada sistem ini diasumsikan jika Sekretaris DPU yang bertugas untuk mencetak semua permohonan yang telah disetujui oleh otorisator berstatus sebagai karyawan <i>outsourcing</i> . Sehingga ia hanya dapat menerima dan melihat permohonan yang telah disetujui dan mengunduhnya.
4.	Pemakaian API untuk menghubungkan perangkat lunak dengan basisdata pusat PT. Pupuk Kaltim.	Pembuatan API berada di luar lingkup pekerjaan penelitian ini.

Dari Tabel 4.5 dapat disimpulkan bahwa adanya dokumen SRS sekali pun tidak menjamin perangkat lunak yang dihasilkan dapat memenuhi semua kebutuhan pelanggan. Banyak faktor yang mempengaruhi kesalahan dalam mengembangkan perangkat lunak, salah satunya karena kurangnya kemampuan analisis dalam memahami kebutuhan sistem yang diinginkan pelanggan dan merancang sistem berdasarkan spesifikasi yang ada. Namun terlepas dari kendala tersebut, terdapat beberapa keuntungan dengan adanya dokumen SRS terhadap proses pengembangan perangkat lunak maupun perangkat lunaknya itu sendiri. Berikut keuntungannya:

- a. Membantu pengembang (analisis, *programmer*, *tester*) memahami semua kebutuhan perangkat lunak yang diinginkan pelanggan.
- b. Membantu proses pengembangan perangkat lunak pada fase berikutnya (desain, implementasi, pengujian).
- c. Membantu pekerjaan pengembang menjadi lebih terfokus.
- d. Membantu pengembang mengingat semua spesifikasi yang dimiliki perangkat lunak.
- e. Membantu proses pengecekan seluruh fungsionalitas yang harus dimiliki perangkat lunak guna memastikan fungsionalitas tersebut telah tersedia dan dapat bekerja sesuai dengan harapan.

Selain keuntungan di atas, mengaplikasikan dokumen SRS pada proses pengembangan perangkat lunak juga memiliki kekurangan, diantaranya adalah:

- a. Butuh banyak waktu untuk membuat dokumennya sehingga sulit diaplikasikan jika waktu pengembangan perangkat lunak terbatas. Lalu apabila pihak pembuat dokumen tidak memiliki pengalaman dalam membuat dokumen SRS, dibutuhkan waktu untuk mempelajari cara membuat dokumen ini baik dari format IEEE 830-1998, MIL-STD-498, ataupun format lainnya.

- b. Sulit diterapkan jika pihak pembuat dokumen tidak benar-benar memahami apa yang dibutuhkan oleh pelanggan. Akibatnya dokumen yang dihasilkan akan berubah terus-menerus dan dampaknya proses pengembangan perangkat lunak menjadi lebih lama serta produk yang dihasilkan tidak bisa selesai tepat pada waktunya.

4.2 Pengujian Dokumen SRS

Tabel 4.6 merupakan daftar kekurangan dokumen SRS yang ditemui pihak ahli setelah pengujian dokumen dilakukan. Pihak ahli yang terlibat dalam pengujian ini sebanyak 2 orang dan masing-masing memiliki latar belakang di bidang rekayasa perangkat lunak. Pengujian dilakukan menggunakan kuisioner, di mana pihak ahli menjawab kuisioner tersebut dengan melakukan review teknis terhadap kedua dokumen yang diuji. Dari pengujian yang telah dilakukan ditemukan 16 poin kekurangan pada dokumen SRS di mana 9 poin dilakukan perbaikan dan 7 poin sisanya menjadi catatan tersendiri untuk dikaji lebih dalam pada penelitian selanjutnya. Kekurangan yang telah diperbaiki diberi tanda *check* (✓) dan yang belum dapat diperbaiki diberikan tanda silang (✗).

Tabel 4.6 Daftar Kekurangan Dokumen SRS Beserta Umpan Baliknya

No.	MIL-STD-498	IEEE 830-1998
1.	Detail kebutuhan kualitas perangkat lunak masih kurang tepat. (<i>portability</i> dan <i>reusability</i>)	Informasi terkait kebutuhan fungsionalitas muncul berkali-kali (<i>redundan</i>). ✓
2.	Alasan antarmuka internal perangkat lunak diserahkan kepada <i>programmer</i> .	<i>Leveling</i> kebutuhan (kebutuhan prioritas) belum ada. ✗
3.	Tidak ada subbab 3.3.x (a, b, c, d, e) seperti di panduan.	Tidak ada rincian pada subbab perspektif produk (subbab 2.1) seperti di panduan. ✗
4.	Kebutuhan terkait keamanan (subbab 3.7) seharusnya diletakkan pada subbab persyaratan keamanan dan privasi (subbab 3.8). ✓	Tidak ada rincian pada subbab batasan (subbab 2.4) seperti di panduan. ✗
5.	Tidak ada penjelasan terkait standar keamanan IT di PT. Pupuk Kaltim.	Tidak ada penjelasan mengenai cara koneksi perangkat lunak ke basisdata karyawan, apakah dengan API atau koneksi langsung (LDAP). ✓
6.	Tidak ada spesifikasi kebutuhan terkait perangkat keras komputer (subbab 3.10.1) ✓	
Kesalahan/masukan di kedua dokumen		
7.	Karakteristik antarmuka eksternal tidak lengkap/detil. Bentuk/format antarmuka eksternal akan seperti apa? Pakai metode apa, API / LDAP?	✓
8.	Tidak ada penjelasan terkait penanganan masalah dengan entitas luar, seperti <i>basisdata down</i> , dsb.	✓

9.	Tidak tersedia notasi diagram seperti usecase diagram, UML, ERD, dsb.	✗
10.	Tidak terdapat penomoran unik disejumlah kebutuhan.	✗
11.	<p>Detil standar pengembangan di organisasi kurang digali (yang disebutkan hanya aturan bahasa pemrogramannya saja). Misalnya:</p> <p>a. Lingkungan tempat sistem akan dideploy seperti apa? <i>Shared/dedicated</i>?</p> <p>b. Adakah batasan utilitas hardware? Maksimum CPU / memori?</p> <p>c. Terkait isu privasi dan keamanan maka autentikasi pengguna akan seperti apa? (SOP login/autentikasi)</p>	✓

Berikut ini merupakan pertimbangan atau alasan yang mendasari sejumlah kekurangan belum dapat diperbaiki, yaitu:

a. Alasan antarmuka internal perangkat lunak diserahkan kepada *programmer*

Antarmuka internal perangkat lunak merupakan hubungan antara perangkat lunak dengan berbagai komponen perangkat lunak seperti *library*, *modul*, *package*, *plug-in*, dsb. Oleh sebab itu Analis menyerahkan kebutuhan ini kepada *programmer* karena Analis tidak berwenang untuk menentukan komponen apa yang harus mereka gunakan saat mengembangkan perangkat lunak. Hal itu sepenuhnya menjadi otoritas *programmer* karena terkait dengan kemampuan teknis dan preferensi yang mereka miliki.

b. Tidak ada subbab 3.3.x (a, b, c, d, e) seperti di panduan

Dokumen SRS versi MIL-STD-498 memang tidak memuat subbab 3.3.x (a, b, c, d, e, f). Ada beberapa alasan yang mendasari hal tersebut:

1. Penjelasan pada dokumen panduan terkait subbab ini sulit dipahami. Diperlukan contoh yang lebih banyak dan detail agar analis dapat menangkap maksud dari subbab tersebut.
2. Pada penelitian ini analis tidak merasa informasi pada subbab 3.3.x penting untuk dimuat, oleh karena itu subbab ini diputuskan untuk tidak dibuat.

c. Tidak terdapat leveling kebutuhan (kebutuhan prioritas)

Semua kebutuhan yang dinyatakan pada kedua dokumen tersebut memiliki prioritas yang sama dan harus terpenuhi semuanya. Oleh sebab itu informasi terkait leveling kebutuhan dirasa tidak perlu dinyatakan dikedua dokumen ini.

d. Tidak ada rincian pada subbab perspektif produk (subbab 2.1) seperti di panduan

Pada dokumen SRS versi IEEE rincian terkait subbab perspektif produk sebenarnya sudah ada, hanya saja rincian tersebut tidak diletakkan pada subbab 2.1. Hal itu dikarenakan dokumen ini mengikuti struktur *template* berdasarkan fitur, di mana pada *template* tersebut rincian terkait perspektif produk seperti kebutuhan antarmuka

pengguna, perangkat keras, dan komunikasi tidak lagi diletakkan pada subbab 2.1 melainkan dimuat pada subbab 3.1.

- e. Tidak ada rincian pada subbab batasan (subbab 2.4) seperti di panduan

Tidak semua rincian pada subbab batasan dimuat oleh analis pada dokumen SRS versi IEEE. Analis hanya memuat rincian batasan mengenai kebijakan peraturan (poin a pada panduan). Beberapa alasan yang melatarbelakangi hal tersebut adalah:

1. Pihak pelanggan lebih banyak menyampaikan batasan dari sisi kebijakan peraturan, seperti siapa saja pihak yang dapat menggunakan layanan atau bagaimana alur otorisasi permohonan saat ini.
2. Pelanggan tidak pernah menyampaikan secara detail terkait batasan lain yang harus dimiliki perangkat lunak seperti bagaimana fungsi audit perangkat lunak, kebutuhan terkait *signal handshake protocols* seperti apa dan lainnya.

- f. Tidak tersedia notasi diagram seperti *usecase* diagram, UML, ERD, dsb.

Dokumen ini harus dapat dibaca oleh semua pihak yang terlibat dalam pengembangan perangkat lunak. Karena tidak semua dari mereka memiliki latar belakang dibidang IT, dikhawatirkan penggunaan notasi diagram hanya akan membuat mereka bingung dan kesulitan memahami maksud dari diagram-diagram tersebut. Oleh sebab itu dibuat notasi lain yang lebih sederhana dan mudah untuk mereka pahami seperti pemakaian tabel untuk menjelaskan siapa saja pihak yang dapat menggunakan fungsionalitas pada perangkat lunak (pengganti *usecase diagram*) dan menampilkan hubungan perangkat lunak dengan entitas lain menggunakan diagram sistem secara umum.

- g. Tidak terdapat penomoran unik di sejumlah kebutuhan

Setiap kebutuhan sudah dirangkum ke dalam tabel pelacakan kebutuhan dan disertai dengan penomorannya. Sehingga jika kebutuhan yang ada di kedua dokumen ini ingin dikutip pada dokumen lain maka cukup dengan mengacu pada penomoran yang terdapat pada tabel pelacakan kebutuhan.

Dari sisi kebenaran menurut pihak ahli semua informasi atau kebutuhan yang dimuat di kedua dokumen SRS ini sudah benar dan setiap kebutuhan juga telah dinyatakan secara terukur sehingga dapat diverifikasi kebenarannya. Selain itu kedua dokumen juga sudah cukup untuk memenuhi standar penulisan dokumen SRS. Akan tetapi dari sisi kelengkapan, karena di kedua dokumen ini tidak tersedia notasi diagram seperti *usecase*, UML, dan sebagainya, alhasil pihak ahli menganggap jika dokumen ini belum lengkap. Beberapa penjelasan di sejumlah bagian dokumen juga masih dirasa ambigu atau belum jelas namun

kekurangan ini telah diperbaiki pada versi dokumen terbaru. Sedangkan dari faktor keringkasan, kedua dokumen sudah cukup ringkas dan jelas, namun karena belum lengkap maka dokumen ini belum memenuhi parameter keringkasan dan kelengkapan.

4.3 Evaluasi Hasil Komparasi

Pengujian pada kedua dokumen ini tidak hanya dilakukan oleh pihak dosen Rekayasa Perangkat Lunak namun juga mahasiswa Teknik Informatika Universitas Islam Indonesia. Masing-masing pihak berjumlah 2 orang. Pada awalnya pihak mahasiswa ditujukan untuk melakukan *usability testing*, namun karena banyaknya dokumen yang harus diuji sehingga hasil pengujiannya kurang memuaskan, alhasil pengujian tersebut diputuskan untuk tidak dilanjutkan.

Walau masih ditemui adanya kekurangan di kedua dokumen tersebut, namun menurut pihak ahli secara umum kedua dokumen itu sudah memenuhi standar penulisan dokumen SRS. Selain itu mereka juga menyatakan bahwa dokumen SRS versi IEEE lebih lengkap dan mudah untuk dipahami daripada dokumen SRS versi MIL-STD-498.

Pengujian yang dilakukan kepada mahasiswa juga menunjukkan hasil yang sama, mereka mengatakan bahwa dokumen SRS versi IEEE lebih lengkap, di mana pada dokumen tersebut sudah tersedia prototipe antarmuka dan skenario dari masing-masing fungsionalitas, sehingga mereka mampu memiliki gambaran mengenai sistem yang akan dikembangkan. Selain itu pihak mahasiswa juga menyatakan bahwa dokumen SRS versi IEEE dan dokumen panduannya lebih nyaman untuk dibaca, dan mudah dipahami. Hanya saja dokumen yang dihasilkan oleh format ini terlalu tebal sehingga lebih banyak waktu yang diperlukan untuk membacanya.

Sebaliknya, menurut para mahasiswa, dokumen SRS versi MIL-STD-498 jauh lebih ringkas. Hal itu menjadi nilai positif dari standar ini di mata mereka. Akan tetapi dokumen panduan yang tersedia cenderung sulit untuk dipahami dan dokumen SRS tersebut kurang lengkap sehingga tidak dapat memberikan mereka bayangan mengenai sistem yang akan dikembangkan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- a. Telah berhasil dilakukan komparasi terhadap standar pembuatan dan dokumen SRS dari MIL-STD-498 dan IEEE 830-1998.
- b. Hasil komparasi menunjukkan sejumlah perbedaan pada standar MIL-STD-498 dan IEEE 830-1998 baik dari sisi struktur dokumen, subjek bahasan, maupun beberapa istilah yang digunakan. Perbedaan tersebut kemudian mempengaruhi faktor keringkasan dan kelengkapan dari dokumen SRS yang dihasilkan, terutama perbedaan pada struktur dokumen dan subjek bahasan.
- c. Bagi pengembang pemula sangat direkomendasikan menggunakan IEEE 830-1998 ketika membuat dokumen SRS karena pertimbangan berikut:
 1. Dari sisi keringkasan, dokumen SRS versi MIL-STD-498 memang lebih unggul karena jumlah halaman yang dihasilkan tidak sebanyak dokumen SRS versi IEEE. Namun dari sisi kelengkapan, dokumen SRS versi IEEE lebih lengkap, mudah dimengerti, dan nyaman untuk dibaca meskipun dokumen yang dihasilkan lebih banyak daripada versi MIL-STD-498.
 2. Kelebihan lain yang dimiliki oleh IEEE yaitu panduan yang disediakan oleh *best practice* ini lebih mudah dipahami daripada panduan dari MIL-STD-498. Selain itu referensi terkait IEEE 830-1998 juga tidak sulit untuk dicari, baik yang menggunakan Bahasa Indonesia maupun Bahasa Inggris.

5.2 Saran

Adapun saran yang dapat diberikan dari hasil penelitian yang telah dilakukan yaitu:

- a. Beberapa kekurangan yang ditemui oleh pihak ahli namun belum dapat diperbaiki oleh penulis saat ini (lihat Tabel 4.6) dapat dikaji kembali pada penelitian berikutnya.
- b. Komparasi dapat dilakukan dengan menggunakan standar pembuatan dokumen SRS lain seperti ISO/IEC/IEEE 29148:2011 atau dengan versi yang berbeda contohnya DI-IPSC-81433A (MIL-STD-498) sehingga memungkinkan terjadinya perubahan pada hasil komparasi.

- c. Selain dilakukan review teknis dari para ahli, perlu dipertimbangkan kembali penggunaan *usability testing* untuk menguji dokumen SRS yang telah dihasilkan. Sebab, dengan metode pengujian tersebut kita bisa mendapatkan lebih banyak masukan dari beragam pihak, mulai dari pihak ahli maupun pengembang pemula terkait seperti apa dokumen SRS yang telah dihasilkan.
- d. Perlu dirumuskan kembali skenario pengujian *usability* yang lebih efektif guna menguji dokumentasi teknis khususnya dokumen SRS agar pihak partisipan tidak merasa jenuh dan lelah sewaktu melakukan pengujian dan kualitas jawaban yang diberikan pun tidak mengecewakan.
- e. Perlu dibuat *template* dokumen SRS versi baru dengan menggabungkan *template* dari MIL-STD-498 dan IEEE 830-1998 sehingga dokumen SRS yang dihasilkan dapat memenuhi kualitas yang lebih ringkas, lengkap dan mudah diterapkan oleh pengembang pemula.

DAFTAR PUSTAKA

- Ali, M. J. (2006). *Metrics for Requirements*. Umea University, Sweden.
- Arthur, J. D., & Stevens, K. T. (1992). Document quality indicators: A framework for assessing documentation adequacy. *Journal of Software Maintenance: Research and Practice*. <https://doi.org/10.1002/smr.4360040303>
- Collins, M. (2010). Implementing Usability Testing Of Technical Documents At Any Company And On Any Budget.
- Departemen of Defense. (1994). *Military Standard Software Development and Documentation*. Washington, DC: Departemen of Defense United States of America.
- Ganesh, S., & Gunda. (2008). *Requirements Engineering: Elicitation Techniques*. University West, Departement Teknologi, Matematika, dan Ilmu Komputer. Trollhättan: University West.
- Graham, D., Veenendaal, E. v., Evans, I., & Black, R. (2008). *Foundations of Software Testing: ISTQB Certification*. Cengage Learning EMEA.
- Hedwing, R. (2011). *Teori Sistem - BINUS University*. Dipetik Mei 16, 2017, dari binus.ac.id: binus.ac.id/wp-content/uploads/2011/11/Teori-Sistem1.pdf
- IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. New York: IEEE Xplore.
- Krug, S. (2000). *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. Amerika Serikat: New Riders Press.
- Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements: A Use Case Approach, Second Edition*. Addison Wesley.
- Nugraha, O. I., & Afandi, K. (2016). *Rancang Bangun Sistem Informasi Job Order Request Berbasis Web Dengan Menggunakan Aplikasi Joget Workflow*. Bontang.
- Nordquist, R. (2017, April 6). *Comparison - Glossary of Grammatical and Rhetorical Terms - Definition and Examples*. Dipetik Agustus 15, 2017, dari ThoughtCo.: <https://www.thoughtco.com/comparison-composition-and-rhetoric-1689882>
- Pressman, R. S. (2009). *Software Engineering: A Practioner's Approach*. <https://doi.org/10.2991/978-94-6239-006-5>

- Requirement Management School. (2010, Juli 1). *Requirement Validation*. Dipetik Mei 16, 2017, dari Requirement Management School: http://www.requirementsmanagement.school.com/w1/Requirements_Validation
- Rouse, M. (2016, Oktober). *Definition Software Development*. Dipetik Mei 3, 2017, dari WhatIs.com: <http://whatIs.techtarget.com/definition/software-development>
- Tullis, T., & Albert, B. (2008). *Measuring The User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann.
- Vie, D. L., Jr. (2007). *Writing Software Requirements Spesification*. Dipetik April 17, 2017, dari www.letu.edu: <http://www.letu.edu/people/jaytevis/Software-Engineering/Examples/writing-software-requirements-specifications.doc>
- Wahono, R. S. (2006, April 29). *Menyegarkan Kembali Pemahaman Tentang Requirements Engineering*. Dipetik Mei 8, 2017, dari RomiSatriaWahono.net: <http://romisatriawahono.net/2006/04/29/menyegarkan-kembali-pemahaman-tentang-requirement-engineering/>
- Westfall, L. (2006). *Software Requirements Engineering: What, Why, Who, When, and How* By Linda Westfall. *ASQs Software Quality Professional Journal*, (2004), 9–15.

LAMPIRAN

Lampiran A: Kuisisioner pengujian dokumen SRS untuk pihak Ahli.

**Kuesioner Penelitian Komparasi Kualitas Standar
Dokumen SRS: Standar Militer MIL-STD-498 dan IEEE 830-1998
(Studi Kasus : Perangkat Lunak Job Order Request PT. Pupuk Kaltim)**

Petunjuk Pengisian

1. Pengujian **hanya** dilakukan pada dokumen SRS yang dihasilkan dari standar MIL-STD-498 dan IEEE 830-1998. Bukan pada panduannya.
2. Silahkan isi data di bawah ini.
 Nama penguji : _____
 Tanggal pengujian dilakukan : _____
3. Terima kasih atas waktu yang telah diberikan.

1. Dalam membuat dokumen SRS manakah panduan yang paling sering anda gunakan atau familier menurut anda?

☐ IEEE 830-1998

☐ MIL-STD-498

☐ IEEE 1233

☐ Lainnya, _____.

No	Pertanyaan	MIL-STD-498		IEEE 830-1998	
		Ya	Tidak	Ya	Tidak
2.	Apakah anda tahu atau pernah mendengar panduan pembuatan dokumen SRS MIL-STD-498 dan IEEE 830-1998 sebelum terlibat dalam pengujian ini?				
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?				
4.	Semua informasi/kebutuhan di dokumen SRS sudah dimuat pada subbab yang tepat.				
5.	Semua kebutuhan di dokumen SRS sudah menggunakan pernyataan yang terukur sehingga dapat diverifikasi.				
6.	Semua istilah, informasi, atau kebutuhan sudah dinyatakan dengan jelas, mudah dipahami, dan				

	tidak ambigu.				
7.	Dokumen SRS ini sudah ringkas. Artinya informasi/kebutuhan terkait perangkat lunak sudah dijelaskan 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.				
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.				
10.	Kedua dokumen SRS ini sudah cukup memenuhi standar penulisan dokumen SRS.				
11.	Subbab Kebutuhan Kemampuan Perangkat Lunak dan Fungsionalitas (subbab 2.3) telah disusun dengan benar.				

12. Kekurangan apa yang anda temui pada subbab 2.3 di dokumen SRS versi MIL-STD-498?

- ☐ Penjelasan yang membingungkan beberapa informasi/kebutuhan.
- ☐ Terdapat informasi/kebutuhan yang muncul berulang-ulang (*redundant*).
- ☐ Terdapat informasi/kebutuhan yang tidak konsisten dengan informasi/kebutuhan pada dokumen SRS versi IEEE.
- ☐ Terdapat informasi/kebutuhan yang tidak penting dan seharusnya tidak perlu dimuat pada dokumen SRS.
- ☐ Tidak memiliki skenario masukan/keluaran disetiap kemampuan perangkat lunak seperti IEEE.
- ☐ Lainnya

13. Kekurangan apa yang anda temui pada subbab 2.3 di dokumen SRS versi IEEE 830-1998?

- ☐ Deskripsi yang membingungkan di beberapa fungsionalitas perangkat lunak.
- ☐ Terdapat informasi/kebutuhan yang muncul berulang-ulang.
- ☐ Terdapat informasi/kebutuhan yang tidak konsisten dengan informasi/kebutuhan pada dokumen SRS versi MIL-STD-498.
- ☐ Terdapat informasi/kebutuhan yang tidak penting dan seharusnya tidak perlu dimuat pada dokumen SRS.
- ☐ Lainnya

14. Secara umum kedua dokumen ini masih memiliki sejumlah kekurangan, diantaranya:

- ☐ Tidak tersedia notasi diagram seperti *usecase* diagram, UML, ERD, dsb.
- ☐ Beberapa kebutuhan bukan merupakan informasi penting yang harus dimuat pada dokumen.
- ☐ Sejumlah kebutuhan tidak diberi penomoran yang unik, sehingga sulit dilakukan pengujian
- ☐ Kebutuhan yang sama muncul secara berulang-ulang (*redundant*).
- ☐ Lainnya:

15. Dari kedua dokumen SRS, manakah dokumen yang memberikan informasi lebih rinci serta lebih mudah anda pahami?

- ☐ MIL-STD-498 ☐ IEEE 830-1998