

Software Requirement Specification

Dokumen Software Requirement Specification (SRS) merupakan sebuah penjelasan tentang cara pengembangan dari sebuah software. Secara sederhana, SRS berisikan roadmap tentang semua pihak yang terlibat dalam sebuah proyek development software. Dokumen ini menawarkan spesifikasi fungsional dan non-fungsional dari perangkat lunak dan juga mencakup use cases yang menggambarkan interaksi pengguna yang berada di dalam sistem dari awal hingga akhir.

Minggu 10

Software Requirement Specification

1.1 Pendahuluan

Dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) berfungsi sebagai dasar bagi setiap proyek perangkat lunak yang sukses, yang merinci persyaratan penting, fungsi, dan batasan yang diperlukan untuk memenuhi harapan para pemangku kepentingan. Dalam pengembangan perangkat lunak, persyaratan yang jelas, terdefinisi dengan baik, dan terdokumentasi secara menyeluruh sangat penting untuk menghindari kesalahan yang merugikan dan memastikan keselarasan antar tim.

SRS berfungsi sebagai cetak biru komprehensif yang menguraikan setiap aspek perilaku, kinerja, dan kegunaan perangkat lunak yang diinginkan. Dengan mendefinisikan elemen-elemen ini sejak awal, SRS meminimalkan risiko pengembangan, mencegah perluasan cakupan, dan memastikan jalur yang lebih lancar dari konsep hingga penyelesaian. Jika dilakukan dengan benar, dokumen SRS memperlancar komunikasi antara pengembang, manajer proyek, dan klien, menciptakan visi terpadu untuk proyek dan menyiapkan panggung untuk keberhasilan jangka panjang.

1.2 Apa itu Dokumen SRS?

Dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) adalah deskripsi terperinci dan terstruktur tentang persyaratan fungsional dan non-fungsional suatu sistem perangkat lunak. Berfungsi sebagai panduan definitif bagi pengembang, perancang, dan pemangku kepentingan, SRS menguraikan secara tepat apa yang harus dilakukan perangkat lunak untuk memenuhi kebutuhan bisnis dan pengguna. Dengan mencakup aspek teknis dan operasional, SRS memastikan semua pihak yang terlibat memiliki pemahaman yang sama tentang tujuan dan cakupan proyek.

SRS menonjol dari dokumen persyaratan lainnya, seperti Dokumen Persyaratan Bisnis (BRD) atau Dokumen Spesifikasi Fungsional (FSD), dengan menawarkan pandangan teknis yang lengkap dari keduanya **apa** sistem akan melakukan dan **bagaimana** akan beroperasi. Tidak seperti BRD, yang terutama menggambarkan sasaran bisnis tingkat tinggi, SRS membahas spesifikasi teknis terperinci, termasuk persyaratan fungsional, tolok ukur kinerja, kebutuhan keamanan, dan interaksi sistem.

Tujuan Utama SRS meliputi:

1. **Mendefinisikan Ruang Lingkup Proyek:** Menetapkan batasan proyek secara jelas, mengurangi ambiguitas dan mencegah perluasan cakupan.

2. **Menetapkan Keselarasan Proyek:** Menyelaraskan semua pemangku kepentingan, memastikan bahwa tim pengembangan, manajer proyek, dan pengguna akhir memiliki harapan yang konsisten.
3. **Menyediakan Dasar untuk Validasi dan Pengujian:** Bertindak sebagai tolok ukur untuk memvalidasi produk akhir terhadap persyaratan yang telah ditetapkan sebelumnya, mendukung jaminan kualitas dan memastikan perangkat lunak yang dikirimkan memenuhi tujuan yang dimaksudkan.

Dengan membedakan dirinya sebagai dokumen persyaratan yang komprehensif, SRS menjadi sangat berharga dalam memandu proses pengembangan, meminimalkan risiko proyek, dan menetapkan jalur yang jelas dari perencanaan proyek hingga penyelesaian.

1.3 Pentingnya Sebuah Dokumen Software Requirement Specification (SRS)

Menulis spesifikasi kebutuhan perangkat lunak sangat penting untuk proses pengembangan yang terstruktur dengan baik.

Berikut ini adalah alasannya.

Konsistensi dan kejelasan

SRS mendefinisikan setiap detail di awal sehingga semua orang memahami tujuan proyek. Tanpa itu, prioritas bisa tidak selaras, yang mengarah ke produk akhir yang terputus-putus.

Contoh: Tanpa SRS, beberapa pengembang mungkin akan berfokus pada perancangan antarmuka yang bersih dan ramah pengguna, sementara yang lain akan memprioritaskan fitur backend yang kompleks seperti pemrosesan data. Tanpa prioritas yang disepakati, produk dapat menjadi terputus-putus dan gagal memenuhi kebutuhan pengguna. SRS mencegah hal ini dan memastikan upaya semua orang selaras.

Komunikasi yang ditingkatkan

SRS mendorong komunikasi yang efektif dan merupakan titik acuan bagi anggota tim teknis dan non-teknis.

SRS menguraikan persyaratan dalam bahasa yang jelas, membantu pemangku kepentingan, seperti manajer proyek atau klien, memahami ruang lingkup proyek-bahkan tanpa latar belakang teknis. Pemahaman bersama meminimalkan kesalahpahaman, menjaga agar umpan balik tetap terfokus, dan memastikan semua tim bekerja selaras.

Contoh: Pertimbangkan sebuah fitur yang dimaksudkan untuk meningkatkan keamanan data untuk aplikasi keuangan. Seorang manajer proyek mungkin menafsirkan 'keamanan data' sebagai membutuhkan otentikasi pengguna, sementara pengembang

mungkin melihatnya sebagai protokol enkripsi. SRS mengklarifikasi persyaratan keamanan tertentu, sehingga setiap anggota tim memahami pendekatan yang dimaksud.

Mengurangi risiko dan penundaan proyek

SRS mengurangi risiko dengan menciptakan jalur yang jelas untuk pengembangan dan menangani potensi masalah sebelum muncul. SRS menyediakan struktur dan titik referensi, membantu tim menavigasi perubahan tanpa mengganggu kemajuan dan menyebabkan ruang lingkup merayap.

Contoh: Seorang klien meminta fitur baru di tengah-tengah pengembangan. Dengan SRS, tim dapat dengan cepat menilai apakah perubahan tersebut sesuai dengan persyaratan yang ditetapkan dan menentukan dampak potensial.

Apabila Anda ingin membuat sebuah aplikasi *chatting* dengan tampilan dan fitur-fitur tertentu dan ingin aplikasi ini ditujukan khusus untuk perusahaan, mungkin Anda akan berpikir untuk menghilangkan beberapa fitur yang digunakan aplikasi *chatting* komersial yang digunakan banyak orang dan hanya berfokus pada fitur yang dibutuhkan oleh internal karyawan perusahaan. Akan tetapi, posisi Anda saat ini adalah bukan orang atau pihak yang punya *concern* terhadap pengembangan sebuah aplikasi. Hal ini menyebabkan Anda mulai berpikir untuk

melakukan *outsource* atau kerja sama dengan pihak ketiga untuk mengembangkan aplikasi *chatting* tersebut.

Masalahnya kemudian adalah Anda perlu memastikan bahwa pihak ketiga tersebut tahu persis tentang tujuan yang ingin Anda capai dan dapat mewujudkan ide Anda membuat sebuah aplikasi *chatting* menjadi kenyataan. Jika Anda hanya berharap semua itu tanpa memberikan detail yang diperlukan pihak ketiga, bisa jadi biaya yang dikeluarkan dan waktu pengerjaan aplikasi menjadi terlalu mahal. Pihak ketiga yang merupakan *software developer* Anda bisa salah langkah dalam mengerjakan terutama dalam aktivitas pemrograman atau *coding* dari aplikasi Anda.

Oleh karena itu, Anda membutuhkan dokumen yang dapat menyampaikan semua gagasan Anda dan mencakup semua detail yang dibutuhkan oleh pihak ketiga. Anda harus menyampaikan ide dan gagasan Anda ke dalam bahasa yang dipahami oleh pihak ketiga.

Software Requirement Specification (SRS) adalah sebuah dokumen yang menjelaskan hal-hal yang diinginkan oleh klien dan hal-hal yang dapat disediakan oleh vendor atau perusahaan yang di *outsource*. Dokumen SRS bisa dianggap sebagai dokumentasi perjanjian tertulis yang mencakup detail aplikasi yang dikerjakan.

Sebagaimana dokumen-dokumen lain yang berisikan penjelasan detail mengenai hal yang akan dikerjakan, dokumen SRS juga dapat membantu pihak ketiga memperkirakan biaya dan

waktu pengerjaan. Dokumen ini juga dapat membantu kepada *developer* dari pihak ketiga untuk menentukan teknologi yang diperlukan untuk menyelesaikan pekerjaan. Bukan hanya itu, beberapa hal ini juga dapat terbantu berkat adanya dokumen SRS:

- UI/UX Desainer dari pihak ketiga juga mendapatkan gambaran tentang proyek secara utuh sehingga dapat menyesuaikan desain dengan *use case* yang tersedia dalam SRS.
- *Software Tester* mendapatkan informasi untuk menyiapkan pengujian terhadap fitur-fitur yang akan dikembangkan dan disesuaikan dengan kebutuhan bisnis klien.
- *End users* dapat memahami tentang gambaran umum *software* yang akan dikerjakan dan hasil yang ingin dicapai.
- Bahkan investor pun bisa mendapat informasi lengkap tentang fitur-fitur yang ada di dalam sistem sehingga bisa membuat keputusan investasi yang tepat.

1.4 Apa Saja Yang Termasuk Di Dalam Dokumen SRS?

Dokumen SRS harus memiliki informasi yang cukup detail untuk pihak ketiga selaku pengembang atau *developer software*. Dokumen ini tidak hanya mendeskripsikan *software* yang akan dibuat, tetapi juga harus memuat informasi mengenai tujuan pembuatan *software* dan

ekspektasi mengenai kinerja seharusnya dari *software* yang akan dihasilkan. Beberapa hal yang harus diperhatikan dalam membuat dokumen SRS adalah tersedianya informasi yang jelas dan tidak ambigu, detail, lengkap, terdapat aspek prioritas pekerjaan, serta ekspektasi output yang jelas.

Biasanya dokumen SRS mencakup hal-hal berikut ini:

- Project overview
- Tujuan dari *software* yang akan dikembangkan
- Deskripsi *software* secara keseluruhan
- Berbagai fungsionalitas yang ada di dalam sistem
- Performa *software* dalam tahap produksi (*production stage*)
- Kebutuhan non-fungsional dalam sistem
- *External interfaces* atau hubungan interaksi *software* dengan perangkat yang lain
- Batasan dari desain atau gambaran yang akan dijalankan oleh sistem

1.5 Komponen Utama Dokumen SRS

Dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) yang efektif disusun untuk memberikan garis besar yang jelas dan komprehensif dari semua persyaratan sistem, memastikan setiap elemen dapat dipahami dan dapat ditindaklanjuti. Berikut adalah rincian komponen-komponen penting:

1. Pengantar

Bagian Pendahuluan menetapkan dasar untuk SRS, merinci isi dokumen **tujuan, cakupan**, dan kritis **terminologi**. Mendefinisikan elemen-elemen ini sejak awal mengurangi ambiguitas dan memastikan pembaca dari berbagai latar belakang teknis memahami tujuan inti proyek.

- **Tujuan:** Menyatakan dengan jelas mengapa perangkat lunak sedang dikembangkan, untuk siapa perangkat lunak itu dikembangkan, dan apa tujuan dokumen tersebut.
- **Cakupan:** Menentukan batasan fungsionalitas perangkat lunak, menetapkan ekspektasi yang jelas tentang apa yang akan dan tidak akan dicakup oleh proyek.
- **Definisi, Akronim, dan Singkatan:** Menyediakan glosarium untuk menstandarisasi istilah dan memperjelas bahasa teknis, mendukung pemahaman yang konsisten di antara para pemangku kepentingan.

2. Deskripsi Keseluruhan

Bagian ini menawarkan pandangan tingkat tinggi tentang perangkat lunak, membantu pembaca memahami konteks sistem, pengguna, dan tujuan.

- **Perspektif Produk:** Menjelaskan bagaimana perangkat lunak cocok dengan sistem yang lebih besar atau berhubungan dengan produk yang ada, termasuk dependensi, antarmuka, atau integrasi.

- **Fitur Produk:** Merangkum fitur-fitur utama, menyediakan ikhtisar fungsional yang menjelaskan kemampuan inti perangkat lunak tanpa membahas detail yang terperinci.
- **Kelas dan Karakteristik Pengguna:** Mengidentifikasi berbagai jenis pengguna akhir, mencatat kebutuhan atau batasan pengguna tertentu untuk memandu desain yang berpusat pada pengguna.

Deskripsi ini memberikan orientasi penting, membantu pembaca memvisualisasikan bagaimana sistem akan berfungsi dalam lingkungannya dan siapa yang akan dilayaninya.

3. Persyaratan Khusus

Bagian Persyaratan Khusus membahas secara terperinci persyaratan fungsional dan non-fungsional, yang menetapkan ekspektasi teknis yang jelas.

- **Persyaratan Fungsional:** Menguraikan tindakan inti yang harus dilakukan perangkat lunak, seperti pemrosesan data, tindakan antarmuka pengguna, atau respons sistem terhadap input tertentu. Setiap persyaratan harus jelas, dapat diuji, dan didokumentasikan dengan contoh atau kasus penggunaan jika berlaku.
- **Persyaratan Non-Fungsional:** Membahas kinerja, keamanan, keandalan, dan kegunaan sistem. Misalnya, dapat menentukan waktu respons, standar perlindungan data, atau kriteria aksesibilitas.

- **Gunakan Kasus:** Skenario terperinci yang menunjukkan bagaimana pengguna akan berinteraksi dengan perangkat lunak, menawarkan wawasan berharga tentang perjalanan pengguna dan perilaku sistem yang diharapkan.

Spesifikasi ini memastikan perangkat lunak memenuhi standar yang ditetapkan dan beroperasi sebagaimana mestinya di berbagai skenario dan interaksi pengguna.

4. Lampiran dan Indeks

Lampiran dan Indeks menyediakan sumber daya tambahan dan navigasi yang mudah:

- **Lampiran:** Sertakan informasi tambahan seperti diagram, model data, atau referensi eksternal yang menambahkan konteks tetapi tidak penting untuk persyaratan inti.
- **Indeks:** Glosarium atau indeks istilah dan singkatan mendukung referensi cepat dan meningkatkan kegunaan dokumen, terutama untuk proyek rumit dengan jargon teknis.

Menggabungkan komponen-komponen terstruktur ini memastikan dokumen SRS tetap jelas, terorganisir, dan komprehensif, memandu pengembangan dari perencanaan awal hingga validasi produk akhir.

1.6 Spesifikasi Persyaratan Perangkat Lunak (SRS) Vs Spesifikasi Persyaratan Bisnis

Orang terkadang mencampurkan konsep perangkat lunak dan spesifikasi kebutuhan bisnis. Sebenarnya keduanya sangat berbeda.

Perbedaan utama antara spesifikasi kebutuhan perangkat lunak dan spesifikasi kebutuhan bisnis adalah bahwa yang pertama menangkap semua informasi yang terkait dengan perangkat lunak sedangkan yang kedua menangkap semua informasi yang terkait dengan bisnis.

Aspek	Spesifikasi Persyaratan Perangkat Lunak (SRS)	Spesifikasi Persyaratan Bisnis (BRS)
Definisi	Dokumen yang menguraikan persyaratan fungsional dan non-fungsional dari sistem perangkat lunak.	Dokumen yang mendefinisikan kebutuhan dan tujuan bisnis tingkat tinggi untuk suatu proyek atau produk.
Tujuan	Menyediakan spesifikasi teknis bagi pengembang untuk membangun perangkat lunak.	Menjelaskan apa yang perlu dicapai bisnis melalui proyek atau produk.
Para penonton	Terutama ditujukan untuk tim pengembangan, QA, dan pemangku kepentingan teknis.	Ditujukan kepada pemangku kepentingan bisnis, manajer proyek, dan analis.
Fokus Konten	Rincian fungsionalitas sistem, kinerja, dan kendala desain.	Berfokus pada sasaran bisnis, tujuan, dan persyaratan tingkat tinggi.

Aspek	Spesifikasi Persyaratan Perangkat Lunak (SRS)	Spesifikasi Persyaratan Bisnis (BRS)
Tingkat Detail	Tingkat detail teknis yang tinggi, menentukan setiap fitur dan perilaku perangkat lunak.	Tingkat tinggi dan luas, berfokus pada "apa" daripada "bagaimana".
Persyaratan Jenis	Persyaratan fungsional, persyaratan non-fungsional, dan batasan sistem.	Persyaratan bisnis, kebutuhan tingkat tinggi, dan tujuan tanpa rincian teknis.
Contoh Persyaratan	Sistem harus mendukung hingga 1,000 pengguna simultan; Waktu pemuatan halaman harus <2 detik.	Perangkat lunak tersebut akan meningkatkan kepuasan pelanggan dengan mengurangi waktu respons hingga 20%.
Cakupan	Terbatas pada aspek teknis perangkat lunak yang akan dibangun.	Luas. Mencakup semua kebutuhan dan harapan bisnis untuk proyek.
Lacak	Sangat dapat dilacak ke fitur tertentu, kasus uji, dan spesifikasi teknis.	Dapat dilacak ke tujuan dan sasaran bisnis, biasanya selaras dengan strategi bisnis.
Kepemilikan	Dimiliki oleh tim teknis, seperti pengembangan, teknik, dan QA.	Dimiliki oleh tim bisnis, seperti tim manajemen proyek dan analisis bisnis.
Frekuensi Revisi	Sering direvisi selama fase pengembangan saat persyaratan disempurnakan.	Direvisi lebih jarang, biasanya hanya dengan perubahan besar dalam tujuan bisnis.
Contoh Dokumen	Dokumen persyaratan sistem, dan spesifikasi persyaratan fungsional.	Kasus bisnis, piagam proyek, dokumen tujuan bisnis.

1.7 Apa Saja Langkah-Langkah Menulis Dokumen SRS yang Efektif?

Menyusun dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) berkualitas tinggi memerlukan pendekatan terstruktur, yang memastikan keakuratan dan keselarasan dari awal hingga akhir. Berikut panduan langkah demi langkahnya:

Kumpulkan Persyaratan

Mengumpulkan persyaratan yang akurat dan relevan merupakan langkah pertama dan paling penting dalam penulisan SRS. Teknik-teknik yang digunakan meliputi:

- **Wawancara dan Survei:** Diskusi langsung dengan pemangku kepentingan atau kelompok pengguna untuk memahami kebutuhan dan harapan.
- **Lokakarya:** Sesi kolaboratif yang mempertemukan para pemangku kepentingan untuk bertukar pikiran, berdiskusi, dan menyempurnakan persyaratan.
- **Observasi dan Analisis Pengguna:** Mengamati pengguna akhir berinteraksi dengan sistem yang ada untuk mengidentifikasi potensi peningkatan atau fungsionalitas penting.
- **prototyping:** Membuat model awal untuk memvalidasi dan menyempurnakan persyaratan berdasarkan masukan pengguna.

Teknik-teknik ini membantu menangkap gambaran lengkap tentang apa yang harus dicapai perangkat lunak, menyediakan landasan yang kuat bagi SRS.

Tentukan Ruang Lingkupnya

Menetapkan cakupan proyek yang jelas dalam SRS sangat penting untuk mengelola ekspektasi dan menghindari perluasan cakupan. Saat menetapkan cakupan:

- **Tetapkan Batasan:** Uraikan dengan jelas apa saja yang akan dicakup proyek dan apa yang tidak, dengan fokus pada fungsionalitas dan batasan perangkat lunak yang dimaksudkan.
- **Mengidentifikasi Kendala** Catat setiap ketergantungan, tenggat waktu, atau keterbatasan sumber daya yang dapat memengaruhi proyek.
- **Mengelola Harapan Pemangku Kepentingan:** Tangani potensi perluasan atau fitur tambahan sejak awal untuk mencegah perubahan tak terduga di kemudian hari dalam proyek.

Ruang lingkup yang ditetapkan dengan baik menjaga proyek tetap pada jalurnya dan memastikan semua pemangku kepentingan memiliki pemahaman bersama tentang batasan pengembangan.

Tulis Pendahuluan

Pendahuluan yang ringkas dan terorganisasi dengan baik sangat penting untuk menentukan corak dokumen SRS. Bagian ini harus mencakup:

- **Maksud dan tujuan:** Nyatakan dengan jelas maksud dokumen dan tujuan keseluruhan proyek perangkat lunak.
- **Pemirsa dan Penggunaan** Tentukan siapa yang akan menggunakan dokumen SRS, seperti pengembang, manajer proyek, atau tim QA.
- **Terminologi:** Berikan definisi untuk istilah teknis, akronim, atau jargon apa pun untuk memastikan semua pembaca memahami kontennya.

Pendahuluan yang disusun dengan baik memberikan landasan yang memandu pembaca melalui sisa dokumen dengan jelas.

Jelaskan Sistem Secara Keseluruhan

Bagian ini harus menawarkan gambaran umum tingkat tinggi tentang sistem, termasuk:

- **Perspektif Sistem:** Jelaskan bagaimana perangkat lunak cocok dengan sistem yang lebih besar atau hubungannya dengan produk dan sistem lain.
- **Fungsi Sistem:** Rangkum fungsionalitas inti yang akan disediakan perangkat lunak, dengan deskripsi umum dan fokus pada operasi utama.

- **Karakteristik Pengguna:** Merinci jenis pengguna yang akan berinteraksi dengan sistem, mencatat kebutuhan atau peran khusus, yang akan memandu persyaratan UI/UX dan aksesibilitas.

Mengikuti praktik terbaik untuk bagian ini memastikan pemangku kepentingan memahami bagaimana sistem akan beroperasi dalam lingkungan yang dituju.

Persyaratan Spesifik Detail

Bagian ini menguraikan persyaratan fungsional dan non-fungsional spesifik, dengan menekankan kejelasan, ketepatan, dan kemampuan pengujian.

- **Persyaratan Fungsional:** Uraikan tindakan, respons, dan perilaku yang diharapkan dari perangkat lunak dalam skenario tertentu. Setiap persyaratan harus tepat, tidak menyisakan ruang untuk ambiguitas.
- **Persyaratan Non-Fungsional:** Tentukan standar kualitas seperti kinerja (misalnya, waktu respons), keamanan (misalnya, perlindungan data), dan kegunaan (misalnya, pedoman aksesibilitas).
- **Hindari Ambiguitas:** Gunakan bahasa dan contoh yang lugas jika memungkinkan untuk mencegah salah tafsir.

Dengan mendokumentasikan persyaratan ini secara jelas, SRS memastikan perangkat lunak akan memenuhi kebutuhan pengguna dan standar sistem.

Tinjau dan Validasi Dokumen SRS

Validasi pemangku kepentingan sangat penting untuk memastikan SRS akurat dan sesuai dengan harapan:

- **Sesi Tinjauan Pemangku Kepentingan** Jadwalkan rapat tinjauan rutin dengan pemangku kepentingan untuk mengonfirmasi persyaratan dan mengklarifikasi setiap poin yang membingungkan.
- **Umpan Balik Loop:** Dorong umpan balik dan buat revisi seperlunya untuk mengatasi kekhawatiran pemangku kepentingan.
- **Lacak** Pastikan bahwa setiap persyaratan dapat ditelusuri kembali ke kebutuhan atau tujuan bisnis tertentu untuk memfasilitasi validasi dan pengujian.

Tinjauan berkala mengurangi risiko ketidakselarasan persyaratan, sehingga proyek tetap pada jalurnya.

Perbarui dan Pertahankan Dokumen SRS

Dokumen SRS harus menjadi dokumen yang terus berkembang seiring dengan kemajuan proyek. Praktik-praktik utama meliputi:

- **Kontrol Versi:** Terapkan versi untuk melacak perubahan dan memelihara catatan versi sebelumnya.
- **Tinjauan Berkelanjutan:** Perbarui dokumen secara berkala untuk mencerminkan setiap perubahan dalam cakupan proyek, persyaratan, atau kendala eksternal.

- **Adaptability** Pastikan SRS tetap dapat beradaptasi, menggabungkan informasi baru atau penyesuaian sesuai tuntutan proyek.

Komitmen untuk menjaga relevansi dokumen SRS sepanjang siklus hidup pengembangan mendukung keberhasilan proyek jangka panjang.

Mengikuti langkah-langkah ini akan membantu menciptakan dokumen SRS yang komprehensif dan berkualitas tinggi yang secara efektif memandu pengembangan perangkat lunak, memastikan kejelasan, keselarasan, dan kemampuan beradaptasi di setiap tahap.

1.8 Kesalahan Umum yang Harus Dihindari Saat Menulis Dokumen SRS

Membuat dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) bisa jadi sulit, dan kesalahan umum sering kali menyebabkan kesalahpahaman, penundaan pengembangan, dan kegagalan mencapai tujuan proyek. Berikut ini adalah beberapa kesalahan utama yang harus dihindari:

1. Menggunakan Bahasa yang Tidak Jelas atau Ambigu Saat Menyusun Dokumen SRS
 - **Kemenduaan:** Istilah yang tidak jelas seperti "cepat", "mudah digunakan", atau "intuitif" dapat disalahartikan. Setiap

persyaratan harus spesifik, terukur, dan bebas dari bahasa subjektif.

- **Istilah Teknis:** Penggunaan istilah teknis secara berlebihan tanpa penjelasan yang jelas dapat membingungkan pemangku kepentingan non-teknis. Sertakan glosarium untuk setiap istilah teknis yang diperlukan guna memastikan kejelasan.

2. Gagal Menyertakan Masukan dari Pemangku Kepentingan

- **Kolaborasi Terbatas:** Tidak melibatkan pemangku kepentingan selama proses berlangsung dapat menyebabkan ekspektasi yang tidak selaras. Sesi umpan balik dan tinjauan rutin dengan semua pemangku kepentingan sangat penting.
- **Mengabaikan Kebutuhan Pengguna:** Mengabaikan persyaratan pengguna akhir atau gagal mengumpulkan masukan pengguna dapat mengakibatkan sistem tidak memenuhi kebutuhan pengguna. Pastikan dokumen SRS mencerminkan permintaan dan skenario pengguna yang sebenarnya.

3. Mengabaikan Persyaratan Non-Fungsional dalam Dokumen SRS

- **Mengabaikan Atribut Kualitas:** Banyak dokumen SRS yang berfokus pada persyaratan fungsional dan mengabaikan aspek non-fungsional seperti kinerja, keamanan, dan

skalabilitas. Mengatasi hal ini sangat penting untuk dokumen yang lengkap.

- **Detail Tidak Memadai:** Persyaratan seperti standar kinerja atau protokol keamanan harus didefinisikan dengan jelas. Uraian yang tidak jelas di sini dapat menimbulkan masalah yang merugikan selama pengembangan.

4. Cakupan yang Tidak Didefinisikan dengan Baik dalam Dokumen SRS

- **Lingkup Creep:** Gagal menetapkan batasan yang jelas akan mengakibatkan cakupan proyek yang terus meluas, yang dapat menyebabkan pembengkakan anggaran dan waktu. Tetapkan apa yang termasuk—dan secara tegas apa yang dikecualikan—sejak awal.
- **Kurangnya Prioritas:** Tidak semua persyaratan memiliki bobot yang sama. Kegagalan dalam menentukan prioritas dapat menyebabkan kebingungan dan salah alokasi sumber daya.

5. Struktur dan Kurangnya Organisasi Dokumen SRS yang Tidak Konsisten

- **Bagian yang Tidak Terorganisir:** Melompat-lompat di antara topik yang tidak berhubungan tanpa struktur yang jelas membuat dokumen sulit dinavigasi. Format yang konsisten dengan bagian-bagian yang logis meningkatkan keterbacaan.
- **Ketertelusuran yang Buruk:** Persyaratan harus dapat dilacak ke tujuan atau kebutuhan pengguna tertentu. Kurangnya

keterlacakan membuat validasi persyaratan dan verifikasi apakah persyaratan telah terpenuhi menjadi lebih sulit.

6. Tidak Memvalidasi atau Meninjau Dokumen SRS

- **Melewati Ulasan:** Terburu-buru dalam proses peninjauan dapat menyebabkan kesalahan yang tidak diperiksa atau persyaratan yang hilang. Sisihkan waktu untuk peninjauan menyeluruh dengan pemangku kepentingan utama.
- **Kriteria Pengujian Tidak Memadai:** Setiap persyaratan harus dapat diuji. Kegagalan dalam menentukan kriteria pengujian atau memasukkan persyaratan yang tidak dapat diverifikasi akan menyebabkan kesulitan dalam fase validasi dan pengujian selanjutnya.

7. Memperlakukan Dokumen SRS sebagai Dokumen Statis

- **Kurangnya Pembaruan:** Persyaratan dapat berkembang, tetapi jika SRS tetap tidak berubah, dokumen tersebut akan cepat menjadi usang. Pertahankan dokumen tersebut sebagai sumber daya yang "hidup", perbarui dokumen tersebut saat tujuan proyek berubah.
- **Tidak Ada Kontrol Versi:** Tanpa versi yang tepat, sulit untuk melacak perubahan atau kembali ke persyaratan sebelumnya. Pastikan semua pembaruan dilacak untuk dokumentasi yang jelas.

Menghindari kesalahan umum ini akan memastikan bahwa dokumen SRS tetap menjadi panduan yang andal, akurat, dan

efektif sepanjang proses pengembangan perangkat lunak, menyelaraskan tujuan proyek dengan kebutuhan pemangku kepentingan dan harapan pengguna.

1.9 Praktik Terbaik untuk Menulis Dokumen SRS yang Efektif

Menulis dokumen Spesifikasi Persyaratan Sistem (SRS) yang efektif adalah kunci untuk memastikan keberhasilan proyek pengembangan perangkat lunak. Berikut adalah beberapa praktik terbaik yang harus diikuti saat menyusun SRS:

- **Jadilah Jelas dan Ringkas**: Tulis persyaratan yang tidak ambigu, sederhana, dan mudah dipahami oleh semua pemangku kepentingan, hindari bahasa yang tidak jelas.
- **Prioritaskan Persyaratan**: Peringkat fitur berdasarkan tingkat kepentingan (harus dimiliki, seharusnya dimiliki, bagus untuk dimiliki) untuk memfokuskan sumber daya pada fungsionalitas kritis.
- **Pastikan Testabilitas**: Tentukan kriteria penerimaan yang terukur untuk setiap persyaratan untuk divalidasi melalui pengujian.
- **Gunakan Alat Bantu Visual**: Sertakan diagram dan diagram alir untuk menjelaskan proses kompleks dan interaksi sistem.

- **Melibatkan Pemangku Kepentingan Secara Berkelanjutan:** Berkolaborasi dengan para pemangku kepentingan sepanjang proyek untuk memastikan keselarasan dan memenuhi kebutuhan yang berkembang.
- **Mencakup Persyaratan Non-Fungsional:** Menangani kinerja, keamanan, skalabilitas, dan kegunaan, beserta persyaratan fungsional.
- **Tetap Perbarui SRS:** Revisi SRS secara berkala seiring proyek perlu berkembang, pastikan ketertelusuran dan kontrol versi yang tepat.

1.10 Perbedaan Antara Spesifikasi Fungsional dan Non-Fungsional

Spesifikasi fungsional adalah hal-hal teknis yang dilakukan di dalam sistem. Spesifikasi ini menjelaskan berbagai fungsi yang dilakukan sistem dalam *software* untuk membantu pekerjaan pengguna (*user*). Sistem fungsional ini juga menentukan respons terhadap masukan atau perintah dari user dan memperhitungkan segala proses yang terjadi di dalamnya. Anda dapat mempertimbangkan sistem fungsional sebagai deskripsi mendetail tentang fitur aplikasi dan kebutuhan penggunanya. Apabila sistem fungsional ini tidak ada dalam dokumen SRS, sama saja dengan meniadakan fungsi dari sistem dalam aplikasi.

Berbeda dengan fungsional, spesifikasi non-fungsional menjelaskan tentang bagaimana sistem itu dapat melakukan sistem fungsional. Spesifikasi ini tidak mempengaruhi fungsionalitas aplikasi. Jika tidak ada spesifikasi ini, sistem tetap akan melakukan perintah atau masukan yang diinginkan oleh penggunanya. Meski demikian, sistem non-fungsional tetap penting untuk dimasukkan ke dalam dokumen SRS karena menentukan karakteristik yang mempengaruhi *user experience*. Alih-alih berfokus pada kebutuhan penggunanya, sistem non-fungsional berfokus pada ekspektasi pengguna untuk meningkatkan kenyamanan saat menggunakan aplikasi. Oleh karena itu, sistem ini biasanya mencakup berbagai hal terkait **performance software**: *aksesibilitas*, keamanan aplikasi, keandalan (*reliability*), response time, testability, privacy, tolerance, documentation, compliance, dan lain sebagainya.

Sebagai contoh sederhana,

- **Teknis kebutuhan fungsional:** Sistem dapat mengirim email ketika kondisi tertentu dipenuhi seperti user sudah subscribe, sign up, mengisi form tertentu
- **Teknis kebutuhan non-fungsional:** Email harus dapat dikirim dengan latency yang tidak lebih besar dari 12 jam

1.11 Bagaimana Menyusun Dokumen SRS?

Cara terbaik untuk menyusun dokumen SRS adalah memulainya dengan membuat kerangka dan informasi umum tentang *software* yang akan Anda kembangkan dan menambahkan detail untuk menyempurnakan draf dokumennya. Berikut adalah enam langkah yang bisa dilakukan untuk menyusun dokumen SRS Anda:

1. Membuat sebuah *outline* dokumen

Langkah paling awal dalam membuat dokumen SRS adalah dengan menyusun kerangka atau *outline* dokumen. Anda dapat menyusunnya dengan kreasi Anda sendiri atau menggunakan *template* dari dokumen SRS sebagai pedoman awal. Berikut ini contoh mendasar dari sebuah kerangka dalam dokumen SRS:

1. Pengantar
2. Latar Belakang Proyek
3. Tujuan Pengembangan
4. Audiens
5. Cakupan Pengembangan
6. Definisi Berbagai Fungsi
7. Deskripsi Keseluruhan
8. Kebutuhan Pengguna
9. Asumsi dan Dependensi
10. Fitur dan Kebutuhan Sistem

- Sistem Fungsional
- Sistem *External Interface*
- Fitur dalam Sistem
- Sistem Non-Fungsional

2. Mendefinisikan Tujuan Anda

Setelah Anda menyusun *outline*, Anda harus menyempurnakannya dengan menentukan tujuan *software* yang akan Anda buat dalam awalan dokumen SRS. Bagian ini merupakan penjelasan mengenai audiens yang dituju dan cara mereka menggunakan *software* yang akan dibuat.

3. Memberikan Gambaran Umum

Bagian berikutnya setelah Anda menentukan tujuan produk yang akan Anda hasilkan adalah merangkum cara kerjanya. Bagian ini merupakan penjelasan mengenai deskripsi yang cukup general tentang fitur di dalam *software* dan cara fitur-fitur tersebut berjalan sesuai dengan kebutuhan penggunaannya. Anda juga perlu menjelaskan tentang asumsi yang Anda buat tentang fungsi dari *software* dan apapun yang bergantung padanya dalam ekosistem teknologi saat ini.

4. Mendeskripsikan Spesifikasi Fungsional dan Non-Fungsional

Sekarang Anda telah sampai pada bagian yang lebih spesifik yakni mendeskripsikan sistem fungsional dan non-fungsional. Penyusunan tiga hal sebelum masuk ke dalam bagian ini membuat Anda memiliki referensi untuk memastikan bahwa segala hal yang

Anda lakukan telah memenuhi kebutuhan dasar dari pengguna *software* Anda. Hal tersebut juga memudahkan Anda dalam mengisi hal yang lebih detail kedepannya, seperti bagian spesifikasi fungsional dan non-fungsional ini.

Bagian ini merupakan deskripsi detail tentang kebutuhan sistem *software* Anda. Hal ini bisa dikatakan sebagai komponen paling penting dalam dokumen SRS Anda. Jelaskan kebutuhan sistem fungsional dengan cukup detail sehingga *developer software* Anda dapat mulai bekerja seperti yang Anda inginkan dan juga jangan lupakan sistem non-fungsional seperti *performance software* dan sekuritasnya.

Dalam bagian ini juga Anda dapat menambahkan *use cases* untuk menjelaskan tentang cara pengguna berinteraksi dengan sistem Anda. Di sinilah Anda dapat mendetailkan tujuan proyek dan mengukur indikator kemajuan proyek selama proses pengembangan *software* Anda berlangsung.

5. Menambahkan Detail Pelengkap

Terakhir dalam penyusunan draft dokumen SRS, Anda perlu menambahkan detail pelengkap sehingga pihak ketiga selaku pengembang perangkat lunak Anda dapat memulai dan menyelesaikan pekerjaan mereka dalam bentuk lampiran, daftar istilah, dan referensi yang juga Anda ketahui.

6. Mengajukan Persetujuan Terhadap Dokumen

Ketika Anda sudah menambahkan berbagai hal detail yang dibutuhkan, kini saatnya meminta persetujuan dari pihak-pihak yang dianggap berwenang (seperti atasan Anda atau jajaran manajerial) terhadap draf dokumen SRS yang telah Anda buat. Kadang dibutuhkan presentasi kepada orang-orang yang terlibat dalam proses *development software* nantinya untuk menjelaskan draf dokumen SRS ini. Tidak menutup kemungkinan akan ada permintaan perubahan atau revisi sehingga Anda harus memperbarui draf dokumen SRS Anda. Tetapi Anda jangan berkecil hati. Ini artinya pihak-pihak yang terlibat, baik dari pengembang *software* Anda dan para pemangku kepentingan di perusahaan Anda, membuat dokumen tersebut lebih tepat sehingga proyek *development software* yang direncanakan tidak akan keluar jalur.

1.12 Kesimpulan

Dokumen Software Requirement Specification (SRS) adalah bagian penting untuk mengerjakan proyek *development software*. Dokumen ini semacam *roadmap* yang memberikan arahan kepada semua pihak yang terlibat dalam proyek sehingga produk yang dihasilkan memenuhi kebutuhan penggunanya. Jika Anda tidak memiliki dokumen SRS yang lengkap sebelum memulai proyek, sulit untuk mengetahui waktu selesainya proyek dan bisa

jadi terjadi pengalihan dengan membuat fitur yang tidak dibutuhkan selama proses pengembangan *software* Anda. Dokumen SRS memberikan Anda perkiraan proyek secara akurat dan menetapkan tugas dari setiap pihak yang terlibat secara efisien.

Penulisan dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) merupakan langkah penting dalam memastikan keberhasilan setiap proyek perangkat lunak. SRS yang terstruktur dengan baik tidak hanya memberikan kejelasan dan arahan bagi tim pengembangan, tetapi juga menyelaraskan harapan pemangku kepentingan, meminimalkan risiko, dan meningkatkan kualitas proyek secara keseluruhan. Dengan menggabungkan komponen-komponen penting, mengikuti praktik terbaik, dan menghindari kesalahan umum, tim dapat membuat dokumen SRS yang efektif yang berfungsi sebagai cetak biru yang andal untuk pengembangan.