Get Press Indonesia

REKAYASA PERANGKAT LUNAK:
KONSEP, METODE, DAN PRAKTIK TERBAIK

Ade Maulana, Nono Heryana, Johni S Pasaribu, Addin Aditya, Elisawati, Rudiansyah, Amna, Angga Aditya Permana, Arief Yanto Rukmana, Rahman Abdillah, Teguh Wahyono

ISBN 978-623-198-658-0

REKAYASA PERANGKAT LUNAK: KONSEP, METODE, DAN PRAKTIK TERBAIK

Ade Maulana
Nono Heryana
Johni S Pasaribu
Addin Aditya
Elisawati
Rudiansyah
Amna
Angga Aditya Permana
Arief Yanto Rukmana
Rahman Abdillah
Teguh Wahyono

GET PRESS INDONESIA

REKAYASA PERANGKAT LUNAK: KONSEP, METODE, DAN PRAKTIK TERBAIK

Penulis :
Ade Maulana
Nono Heryana
Johni S Pasaribu
Addin Aditya
Elisawati
Rudiansyah
Amna
Angga Aditya Permana
Arief Yanto Rukmana
Rahman Abdillah
Teguh Wahyono

ISBN : 978-623-198-658-0

Editor : Diana Purnama Sari,. S.E M.E
Penyunting : Tri Putri Wahyuni,S.Pd
Desain Sampul dan Tata Letak : Atyka Trianisa, S.Pd

Penerbit : GET PRESS INDONESIA
Anggota IKAPI No. 033/SBA/2022

Redaksi :
Jln. Palarik Air Pacah No 26 Kel. Air Pacah
Kec. Koto Tangah Kota Padang Sumatera Barat
Website : www. getpress.co.id
Email : adm.getpress@gmail.com

Cetakan pertama, 2 September 2023

KATA PENGANTAR

Segala Puji dan syukur atas kehadirat Allah SWT dalam segala kesempatan. Sholawat beriring salam dan doa kita sampaikan kepada Nabi Muhammad SAW. Alhamdulillah atas Rahmat dan Karunia-Nya penulis telah menyelesaikan Buku Rekayasa Perangkat Lunak: Konsep, Metode, Dan Praktik Terbaik ini.

Buku ini membahas Pengantar Rekayasa Perangkat Lunak, Manajemen Proyek Perangkat Lunak, Pengembangan Perangkat Lunak Agile, Rekayasa Kebutuhan Perangkat Lunak, Pemodelan Sistem, Arsitektur Perangkat Lunak, Desain dan Implementasi Perangkat Lunak, Pengujian Perangkat Lunak,Evolusi Perangkat Lunak, Penggunaan Ulang Perangkat Lunak, Rekayasa Perangkat Lunak Berbasis Komponen.

Proses penulisan buku ini berhasil diselesaikan atas kerjasama tim penulis. Demi kualitas yang lebih baik dan kepuasan para pembaca, saran dan masukan yang membangun dari pembaca sangat kami harapkan.

Penulis ucapkan terima kasih kepada semua pihak yang telah mendukung dalam penyelesaian buku ini. Terutama pihak yang telah membantu terbitnya buku ini dan telah mempercayakan mendorong, dan menginisiasi terbitnya buku ini. Semoga buku ini dapat bermanfaat bagi masyarakat Indonesia.

Padang, 2 September 2023

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 4
REKAYASA KEBUTUHAN PERANGKAT LUNAK

Oleh Addin Aditya

4.1 Pendahuluan

Seperti yang telah diketahui bersama bahwa dunia ini sudah memasuki era digitalisasi. Perangkat lunak atau software menjadi kebutuhan penting disetiap aspek kehidupan manusia, mulai dari bisnis hingga kehidupan sehari-hari. Kebutuhan yang variatif dari para pemangku jabatan dan individu menuntut adanya pengembangan perangkat lunak yang dapat memenuhi ekspektasi dan kebutuhan mereka. Di satu sisi, kurangnya keterampilan dalam memahami, mengumpulkan, dan mengelola kebutuhan ini dapat menyebabkan proyek perangkat lunak yang kurang memuaskan atau bahkan gagal sepenuhnya.

Untuk mengatasi dan mengantisipasi kegagalan proyek pengembangan perangkat lunak, maka penting untuk memahami konsep rekayasa kebutuhan perangkat lunak baik secara teoritik maupun secara praktik. Rekayasa Kebutuhan adalah disiplin ilmu dan praktik yang berfokus pada proses identifikasi, analisis, dokumentasi, dan manajemen kebutuhan dalam pengembangan perangkat lunak. Proses ini mencakup aktivitas-aktivitas penting seperti pengumpulan kebutuhan dari para stakeholder, analisis untuk mendalami kebutuhan pengembangan perangkat lunak dengan lebih komprehensif, serta pembuatan spesifikasi yang jelas, terstruktur dan tervalidasi.

Urgensi Rekayasa Kebutuhan dalam pengembangan perangkat lunak ialah memastikan bahwa perangkat lunak yang dikembangkan dapat memenuhi ekspektasi pengguna dan memenuhi tujuan bisnis yang mendasarinya. Dengan mengidentifikasi dan mendefinisikan kebutuhan dengan tepat sasaran, maka kegagalan proyek dapat diminimalisir. Rekayasa kebutuhan juga dapat membantu menekan biaya dan waktu karena perubahan yang terlambat dalam siklus pengembangan perangkat lunak.

Dalam implementasinya, Rekayasa Kebutuhan melibatkan berbagai pihak, seperti analis sistem, pengembang perangkat lunak, manajer proyek, pemilik bisnis, dan pengguna akhir. Komunikasi yang efektif dan kerjasama di antara semua pihak ini akan menjadi kunci untuk memahami kebutuhan yang bervariatif dan kompleks.

Rekayasa kebutuhan adalah fase paling efektif dari proses pengembangan perangkat lunak. Fase ini bertujuan untuk mengumpulkan kebutuhan yang baik dari pemangku kepentingan dengan cara yang benar. Penting bagi setiap organisasi untuk mengembangkan produk perangkat lunak berkualitas yang dapat memenuhi kebutuhan pengguna (Pandey, Suman, & Ramani, 2010). Tantangan dalam Rekayasa Kebutuhan Perangkat Lunak meliputi pemahaman mendalam tentang kebutuhan pengguna, manajemen perubahan kebutuhan yang terus berkembang, dan koordinasi yang baik di antara semua pemangku kepentingan. Oleh karena itu, alat dan teknik yang tepat seringkali digunakan untuk membantu dalam proses identifikasi, analisis, dan dokumentasi kebutuhan.

4.2 Pengertian dan Ruang Lingkup Rekayasa Kebutuhan Perangkat Lunak

Rekayasa Kebutuhan Perangkat Lunak adalah suatu disiplin ilmu dan proses yang berfokus pada identifikasi, analisis, dokumentasi, dan manajemen kebutuhan yang berkaitan dengan pengembangan perangkat lunak. Para ahli dalam bidang ini memberikan pengertian yang berbeda, namun pada dasarnya mereka sepakat bahwa Rekayasa Kebutuhan merupakan tahap penting dalam siklus pengembangan perangkat lunak yang bermakna untuk memastikan perangkat lunak yang dikembangkan memenuhi ekspektasi pengguna dan pemangku kepentingan lainnya.

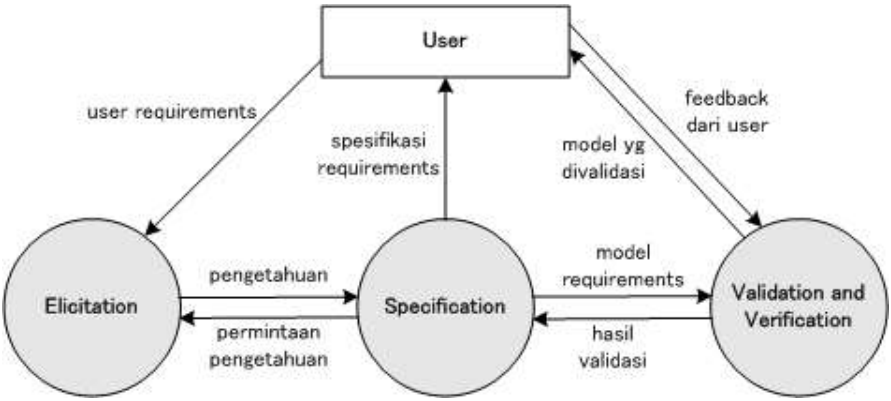
1. Rekayasa kebutuhan adalah serangkaian mekanisme yang tepat untuk memahami apa yang diinginkan pelanggan, menganalisis kebutuhan, menilai kelayakan, menegosiasikan solusi yang masuk akal, menentukan kata kunci dengan jelas, memvalidasi spesifikasi, dan mengelola persyaratan saat diubah menjadi sistem operasional (Pressman, 2010).
2. Rekayasa Kebutuhan adalah proses pemahaman kebutuhan dan kendala sistem yang akan dikembangkan dan menyusun spesifikasi dari kebutuhan-kebutuhan tersebut. Proses ini melibatkan komunikasi yang baik dengan pemangku kepentingan untuk memastikan bahwa kebutuhan mereka dipahami dengan benar dan diimplementasikan dalam perangkat lunak yang dikembangkan (Sommervile, 2011).
3. Rekayasa Kebutuhan adalah proses kolaboratif untuk mendefinisikan fitur-fitur produk yang akan dibangun atau sistem yang akan dikembangkan. Proses ini melibatkan mengumpulkan informasi dari berbagai sumber, memahami kebutuhan pengguna, menetapkan prioritas,

dan menyusun spesifikasi kebutuhan yang jelas dan terukur (Wiegers & Beatty, 2013)

4. Rekayasa kebutuhan merupakan proses penetapan kebutuhan dengan mempelajari kebutuhan pemangku kepentingan dan proses menganalisis dan menyempurnakan spesifikasi tersebut secara sistematis (Hofmann & Lehner, 2001)

Dari pengertian di atas, dapat disimpulkan bahwa Rekayasa Kebutuhan Perangkat Lunak melibatkan proses komprehensif untuk memahami, mendokumentasikan, dan mengelola kebutuhan dari perangkat lunak yang akan dikembangkan. Proses ini berpusat pada pengumpulan data dan informasi yang akurat dari berbagai aktor dan mentransformasikannya menjadi spesifikasi kebutuhan yang jelas dan tervalidasi, sehingga menjadi landasan untuk pengembangan perangkat lunak yang berhasil.

Ruang lingkup Rekayasa Kebutuhan Perangkat Lunak mencakup berbagai aspek yang terkait dengan identifikasi, analisis, dokumentasi, dan manajemen kebutuhan dalam pengembangan perangkat lunak. Ruang lingkup ini melibatkan berbagai tahap dan proses yang bertujuan untuk memastikan bahwa perangkat lunak yang dikembangkan dapat memenuhi harapan dan kebutuhan para pengguna dan pemangku kepentingan lainnya. Berikut adalah ruang lingkup atau cakupan aktivitas dari rekayasa kebutuhan perangkat lunak (Lamsweerde, 2000):



Gambar 4.2. Domain Utama Rekayasa Kebutuhan
Sumber: https://romisatriawahono.net/2006/04/29/menyegarkan-kembali-pemahaman-tentang-requirement-engineering/

1. **Domain Analysis**
   Sistem yang ada saat ini perlu untuk dipelajari terlebih dahulu. Para pemangku kepentingan yang memiliki akses langsung pada sistem perlu untuk diidentifikasi dan diwawancarai guna mendapatkan permasalahan yang terjadi pada sistem saat ini. Selanjutnya akan didapatkan peluang dan target yang akan dibuat.
2. **Elicitation**
   Setelah kebutuhan teridentifikasi, tahap analisis berfokus pada memahami, menganalisis, dan memodelkan kebutuhan tersebut. Analisis kebutuhan bertujuan untuk memastikan bahwa kebutuhan tersebut masuk akal, konsisten, dan dapat diimplementasikan dalam perangkat lunak.
3. **Negotiation and Agreement**
   Setiap alternatif dari kebutuhan pengembangan perangkat lunak akan dievaluasi dan dipertimbangkan berdasarkan analisis risiko dan proses pengambilan keputusan terbaik

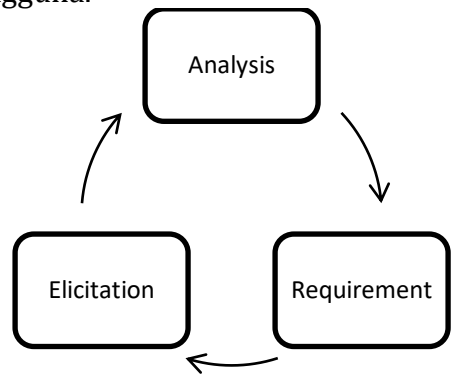
berdasarkan semua aktor yang terlibat dalam proyek pengembangan perangkat lunak.

4. **Verification and Validation**
   Tahap ini melibatkan proses memverifikasi dan memvalidasi kebutuhan untuk memastikan bahwa kebutuhan yang terdokumentasi benar, konsisten, dan memenuhi kriteria kualitas tertentu. Verifikasi kebutuhan berkaitan dengan keakuratan dan kesesuaian kebutuhan dengan kebutuhan pengguna, sementara validasi kebutuhan berfokus pada memastikan bahwa kebutuhan tersebut benar-benar memenuhi kebutuhan bisnis dan pengguna.
5. **Documentation**
   Ruang lingkup ini mencakup proses mendokumentasikan kebutuhan dengan jelas dan terstruktur. Dokumen kebutuhan ini menjadi panduan bagi para pengembang perangkat lunak untuk memahami apa yang harus mereka bangun dan mencapai.
6. **Evolution**
   Ruang lingkup ini mencakup pengelolaan perubahan dan konflik kebutuhan selama siklus hidup pengembangan perangkat lunak. Manajemen kebutuhan memastikan bahwa perubahan kebutuhan dikelola dengan baik dan dipahami oleh semua pemangku kepentingan.

4.3 Requirement Elicitation

4.3.1 Pengertian Elicitation

Requirement elicitation adalah proses pengumpulan, analisis, dan dokumentasi kebutuhan yang diperlukan untuk mengembangkan atau memperbarui suatu sistem atau perangkat lunak. Pengumpulan kebutuhan yang tepat dan komprehensif merupakan langkah penting dalam memastikan bahwa solusi yang dihasilkan akan memenuhi harapan dan kebutuhan pengguna.



Gambar 4.3. Siklus Hidup Requirement Elicitations

Pada gambar 4.2 menunjukkan siklus dari requirement elicitations. Analis akan melakukan kegiatan elisitasi atau pengumpulan kebutuhan, mempelajari atau menganalisis hasil pengumpulan kebutuhan. Analisis yang dimaksud adalah meninjau Kembali hasil pengumpulan kebutuhan, apakah sudah sesuai dengan tujuan atau perlu untuk dilakukan pengumpulan kebutuhan Kembali. Selanjutnya akan ditetapkan kebutuhan yang sah sebagai dasar untuk dikembangkan sebuah proyek perangkat lunak.

4.3.2 Teknik Elicitation

Teknik elisitasi kebutuhan adalah sarana yang digunakan analis sistem untuk menentukan masalah, peluang, dan kebutuhan pelanggan sehingga pengembang sistem dapat membangun sistem yang menyelesaikan masalah tersebut, memanfaatkan peluang tersebut, dan memenuhi kebutuhan pelanggan". Teknik elisitasi adalah alat untuk menemukan & pemahaman yang tepat. Tujuan dari teknik Elisitasi adalah untuk menemukan sebanyak mungkin permasalahan sehingga memudahkan stakeholder untuk mendapatkan aplikasi terbaik yang sesuai dengan kebutuhan (Khan, Dulloo, & Verma, 2014). Berikut adalah beberapa macam teknik elicitation :

1. **Wawancara**
   Wawancara terdiri dari mengajukan pertanyaan kepada pakar atau ahli tentang domain minat dan bagaimana mereka melakukan tugas mereka. Wawancara bisa tidak terstruktur, semi terstruktur, atau terstruktur. Wawancara melibatkan interaksi langsung dengan pemangku kepentingan, pengguna, atau ahli terkait untuk mendapatkan pemahaman mendalam tentang kebutuhan mereka. Wawancara dapat dilakukan secara individu atau dalam kelompok diskusi
2. **Kuesioner**
   Kuesioner adalah teknik yang sangat penting dalam teknik elisitasi kebutuhan, kuesioner membantu mendapatkan informasi dari banyak orang, analis dapat mengumpulkan pendapat dengan dua cara : untuk mendapatkan bukti statistik untuk asumsi, atau untuk mengumpulkan pendapat dan saran.
3. **Observasi**
   Kegiatan ini Melibatkan pengamatan langsung terhadap pengguna atau pemangku kepentingan saat mereka menggunakan sistem yang ada atau dalam konteks yang relevan. Observasi dapat memberikan wawasan tentang

perilaku, kebutuhan yang mungkin tidak disadari, dan tantangan yang dihadapi pengguna (Burge, 2001)

4. **Analisis Dokumen**
   Metode ini adalah metode tidak langsung dan bervariasi tergantung pada dokumen yang tersedia, dan interaksi dengan para ahli. Metode ini melibatkan analisis dokumen yang ada seperti laporan, kebijakan, spesifikasi, atau dokumentasi terkait sistem yang sedang direkayasa. Dokumen ini dapat memberikan pemahaman tentang kebutuhan yang sudah ada, proses bisnis yang ada, atau batasan yang harus diperhatikan.
5. **Prototyping**
   Prototyping telah digunakan untuk teknik elicitation di mana ada banyak ketidakpastian tentang kebutuhan, atau di mana diperlukan *feedback* awal dari pemangku kepentingan (Davis, 1992). Teknik ini melibatkan pembuatan prototipe awal atau model yang digunakan untuk berinteraksi dengan pengguna atau pemangku kepentingan. Prototipe ini membantu dalam memvalidasi dan menggali kebutuhan secara lebih konkret serta memungkinkan pengguna melihat dan merasakan fitur-fitur yang diusulkan.

4.4 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Software Requirement Specifications (SRS) atau Spesifikasi perangkat lunak dapat didefinisikan sebagai pernyataan singkat tentang kebutuhan yang harus dijamin oleh perangkat lunak. Melalui kebutuhan ini, perangkat lunak harus menyediakan fasilitas atau kemampuan kepada pengguna, yang memungkinkan mereka mencapai tujuan organisasi yang ditentukan (Belfo, 2012). Spesifikasi Kebutuhan Perangkat Lunak adalah dokumen yang mendefinisikan kebutuhan

fungsional dan non-fungsional dari suatu sistem perangkat lunak. Dokumen ini menyediakan deskripsi yang lengkap tentang apa yang diharapkan dari sistem yang akan dikembangkan, termasuk fitur-fitur, fungsi, kinerja, batasan, dan persyaratan lainnya.

SRS biasanya dibuat pada tahap awal pengembangan perangkat lunak dan menjadi panduan untuk tim pengembang dalam merancang, mengimplementasikan, dan menguji sistem tersebut. Dokumen ini berfungsi sebagai kontrak antara pengguna atau pemilik proyek dengan tim pengembang, memastikan bahwa semua pihak memiliki pemahaman yang jelas tentang apa yang diharapkan dari sistem yang akan dikembangkan.

4.4.1 Proses Analisis Kebutuhan

Proses analisis kebutuhan dimulai dengan mengidentifikasi dan mendefinisikan masalah atau peluang yang ingin ditangani oleh sistem perangkat lunak. Ini diikuti dengan mengumpulkan persyaratan dari pemangku kepentingan, yang meliputi pengguna, pelanggan, dan pihak lain yang akan terpengaruh oleh sistem perangkat lunak. Persyaratan tersebut kemudian didokumentasikan dan diatur, dan setiap ketidakkonsistenan atau ambiguitas diselesaikan. Langkah terakhir dalam proses analisis persyaratan adalah memvalidasi persyaratan, yang mencakup pengujian dan evaluasi persyaratan untuk memastikan bahwa persyaratan tersebut lengkap, konsisten, dan dapat dicapai

4.4.4 Pemodelan dan Notasi dalam Spesifikasi Kebutuhan

1. **Notasi Pemodelan Proses Bisnis**

   Model dan Notasi Proses Bisnis telah menjadi standar de-facto untuk diagram proses bisnis (Object Management Group, 2011). Notasi Pemodelan Proses Bisnis (BPMN) adalah teknik yang banyak digunakan untuk memodelkan dan menganalisis proses bisnis. Diagram BPMN digunakan untuk merepresentasikan secara visual langkah-langkah

dalam suatu proses, termasuk input, output, dan poin keputusan. Diagram BPMN membantu mengidentifikasi dan mendokumentasikan persyaratan bisnis dan dapat digunakan untuk mengomunikasikan persyaratan tersebut kepada tim pengembangan perangkat lunak

2. **Unified Modelling Language**
   UML (Unified Modeling Language) adalah bahasa yang digunakan untuk mendeskripsikan, memvisualisasikan, membangun, dan mendokumentasikan berbagai artefak dari sistem intensif perangkat lunak (Object Management Group, 2017). Diagram UML dapat digunakan untuk memodelkan berbagai aspek sistem perangkat lunak, termasuk use case, class diagram, dan sequence diagram. Diagram UML berguna untuk memodelkan dan menganalisis persyaratan perangkat lunak dan dapat digunakan untuk mengomunikasikan persyaratan tersebut kepada tim pengembangan perangkat lunak.
3. **Teknik Flowchart**
   Flowchart adalah teknik yang banyak digunakan untuk secara visual mewakili langkah-langkah dalam suatu proses. Flowchart membantu mengidentifikasi dan mendokumentasikan persyaratan dan dapat digunakan untuk mengkomunikasikan persyaratan kepada tim pengembangan perangkat lunak. Flowchart juga membantu dalam mengidentifikasi potensi masalah dan peluang dalam suatu proses.
4. **Diagram Aliran Data (DFD)**
   Data Flow Diagram (DFD) adalah teknik untuk merepresentasikan secara visual aliran data melalui suatu sistem. Diagram DFD berguna untuk mengidentifikasi dan mendokumentasikan persyaratan dan dapat digunakan untuk mengkomunikasikan persyaratan kepada tim pengembangan perangkat lunak. Diagram DFD juga

berguna untuk mengidentifikasi potensi masalah dan peluang dalam suatu sistem.

5. **Gantt-Chart**
   Gantt Charts adalah teknik yang banyak digunakan untuk penjadwalan dan pelacakan kemajuan proyek. Bagan Gantt sangat membantu untuk mengelola dan memantau proses analisis persyaratan, dan dapat digunakan untuk mengkomunikasikan kemajuan proyek kepada pemangku kepentingan
6. **Analisis Kesenjangan**
   Analisis Gap dalam Rekayasa Kebutuhan Perangkat Lunak digunakan untuk mengidentifikasi perbedaan antara keadaan yang ada saat ini dan keadaan yang diharapkan atau yang seharusnya terjadi. Metode ini membantu dalam mengidentifikasi kekurangan, kesenjangan, atau perbedaan antara kebutuhan yang ada dan kebutuhan yang diinginkan. Analisis Gap membantu dalam mengidentifikasi kebutuhan yang hilang, tidak sesuai, atau tidak lengkap dalam pengembangan perangkat lunak. Hal ini memungkinkan tim pengembang untuk mengambil tindakan yang tepat untuk mengurangi kesenjangan dan memastikan bahwa perangkat lunak yang dikembangkan memenuhi kebutuhan yang diinginkan dan diharapkan.

4.5 Verifikasi dan Validasi Kebutuhan

Verifikasi dan validasi kebutuhan perangkat lunak adalah proses untuk memastikan bahwa kebutuhan yang diidentifikasi dalam tahap requirement elicitation telah terverifikasi dengan benar dan valid. Dalam konteks pengembangan perangkat lunak, verifikasi dan validasi kebutuhan penting untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi harapan dan kebutuhan pengguna. Aktivitas verifikasi dilakukan untuk memastikan

bahwa perangkat lunak dikembangkan dengan cara yang benar sementara aktivitas validasi dilakukan untuk memastikan bahwa perangkat lunak dikembangkan sesuai dengan kebutuhan pelanggan (Ponsard, Massonet, Rifaut, & Molderez, 2005).

Verifikasi kebutuhan berfokus pada pemeriksaan apakah kebutuhan yang diidentifikasi telah dijelaskan dengan benar dan apakah kebutuhan tersebut dapat diukur dan diverifikasi. Langkah-langkah verifikasi kebutuhan melibatkan memastikan bahwa setiap kebutuhan telah dituliskan dengan jelas, tidak ambigu, dan spesifik. Selain itu, verifikasi juga melibatkan pengecekan apakah kebutuhan konsisten dan tidak bertentangan satu sama lain.

Validasi kebutuhan berfokus pada memastikan bahwa kebutuhan yang diidentifikasi adalah kebutuhan yang benar-benar diperlukan oleh pengguna dan stakeholder. Proses validasi persyaratan membangun kepercayaan bahwa spesifikasi persyaratan mendefinisikan sistem yang harus memenuhi kebutuhan sebenarnya dari pemangku kepentingan sistem (Lobo & Arthur, 2005). Langkah-langkah validasi kebutuhan melibatkan komunikasi dan konfirmasi dengan stakeholder terkait untuk memastikan bahwa kebutuhan tersebut benar-benar mencerminkan kebutuhan mereka dan menggambarkan masalah yang ingin mereka selesaikan. Ada beberapa model verifikasi dan validasi kebutuhan perangkat lunak yang umum digunakan dalam industri. Berikut ini adalah beberapa model yang sering digunakan :

1. **Model Inspeksi**
   Model ini melibatkan pemeriksaan manual dokumen kebutuhan oleh tim yang terlatih. Tim melakukan analisis dan pengecekan terhadap kebutuhan yang telah diidentifikasi untuk memastikan kejelasan, kesesuaian, dan kebenaran kebutuhan tersebut.
2. **Model Peer-Review**

Dalam model ini, kebutuhan diperiksa oleh rekan sejawat atau anggota tim yang terlibat dalam proyek. Tujuan dari peer review adalah mendapatkan umpan balik dan masukan dari individu yang memiliki pengetahuan dan pengalaman yang relevan.

3. **Model Walkthrough**
   Dalam model walkthrough, kebutuhan disajikan secara formal kepada tim proyek atau kelompok pemangku kepentingan yang relevan. Selama sesi walkthrough, setiap kebutuhan dibahas secara rinci, dan umpan balik dan masukan dari peserta digunakan untuk memperbaiki dan memvalidasi kebutuhan.
4. **Prototyping**
   Dalam model ini, prototipe perangkat lunak dibangun untuk memvalidasi kebutuhan dengan cara memperlihatkan fungsionalitas sistem kepada pemangku kepentingan. Prototipe membantu memvisualisasikan dan menguji fitur-fitur yang diusulkan, dan umpan balik dari pemangku kepentingan digunakan untuk memperbaiki kebutuhan.
5. **Model Uji Kasus**
   Dalam model ini, kebutuhan diuji dengan menggunakan serangkaian kasus uji yang dirancang untuk menguji fungsionalitas dan kinerja perangkat lunak. Kebutuhan dianggap tervalidasi jika dapat memenuhi kasus uji dengan sukses.
6. **Model Formal**
   Dalam model ini, teknik formal seperti logika formal atau model matematika digunakan untuk memverifikasi dan memvalidasi kebutuhan. Metode ini melibatkan analisis formal yang ketat dan dapat membantu menemukan cacat logika atau kelemahan dalam kebutuhan.

4.6 Manajemen Kebutuhan

Memproduksi produk perangkat lunak yang sesuai dengan kebutuhan pemangku kepentingan serta memiliki kualitas yang baik, merupakan tantangan yang paling besar dalam rekayasa kebutuhan perangkat lunak. Buruknya pendefinisian kebutuhan perangkat lunak serta seringnya terjadi perubahan pada kebutuhan tersebut adalah salah satu penyebab jangka waktu pengerjaan proyek perangkat lunak semakin bertambah panjang (Baruah, 2015). Manajemen kebutuhan adalah salah satu kegiatan paling penting untuk keberhasilan pengiriman perangkat lunak dan siklus hidup proyek. Tim pengembangan perangkat lunak yang mengikuti konsep *agile software development* menyambut baik perubahan kebutuhan dalam fase apa pun selama siklus pengembangan perangkat lunak produk. Tabel 4.1 menunjukkan langkah-langkah manajemen kebutuhan dalam berbagai metodologi pengembangan perangkat lunak berbasis *agile.*

Tabel 4.2. Manajemen Kebutuhan Perangkat Lunak berdasarkan Metodologi Pengembangannya (Baruah, 2015)

No	Jenis Metodologi	Deskripsi
1	Extreme Programming	Mendefinisikan kebutuhan berdasarkan *user stories* dan *on-sites customer* atau pelanggan di tempat. *User stories* memiliki dua komponen, yakni **kartu tertulis** dalam dan **percakapan** setelah kartu tertulis. Kartu tertulis hanyalah sebuah '' janji untuk percakapan''. Kartu tertulis tidak perlu lengkap atau

No	Jenis Metodologi	Deskripsi
		dinyatakan dengan jelas. Kartu tertulis boleh dimusnahkan setelah kebutuhan telah didefinisikan
2	Scrum	Metode ini juga mendefinisikan kebutuhan aktual perangkat lunak berdasarkan *user stories.* Product owner memiliki peran penting dalam pengembangan perangkat lunak
3	Feature Driven Development (FDD)	Mengumpulkan kebutuhan pengguna lalu selanjutnya merepresentasikannya ke dalam diagram UML dengan daftar fitur. Daftar fitur mengelola kebutuhan fungsional dan tugas pengembangan. Analisis kebutuhan yang solutif dimulai dengan pemeriksaan tingkat tinggi dari ruang lingkup sistem & konteksnya. Tim menilai domain secara detail untuk setiap area pemodelan. Kelompok kecil menyusun model untuk setiap domain dan mempresentasikan model untuk tinjauan sejawat
4	Lean Software Development	Pengumpulan kebutuhan pengguna dilakukan dengan

No	Jenis Metodologi	Deskripsi
		menghadirkan layar ke pengguna akhir dan mendapatkan masukan mereka. Ideologi produksi *just-in-time* diterapkan untuk mengenali persyaratan dan lingkungan khusus
5	Adaptive Software Development	Pengumpulan kebutuhan dilakukan dalam fase spekulatif. Pertama, menetapkan misi dan tujuan proyek, memahami batasan, menetapkan organisasi proyek, mengidentifikasi dan menguraikan kebutuhan, membuat perkiraan ruang lingkup awal & mengidentifikasi risiko proyek utama
6	Kanban	*User Stories* digunakan untuk membantu memahami apa tujuan sebenarnya dari sebuah sprint. Sprint berisi satu kartu cerita. *User Stories* dipecah menjadi bagian-bagian yang lebih kecil. Sebuah cerita dibagi menjadi tugas sisi klien & sisi server. Tugas dibagi menjadi sub-tugas. Pengembang meminimalkan jumlah item dalam sprint untuk

No	Jenis Metodologi	Deskripsi
		mempertahankan waktu proyek
7	Agile Unified Process (AUP)	Fase kebutuhan termasuk mengidentifikasi pemangku kepentingan, memahami masalah pengguna, menetapkan dasar estimasi & mendefinisikan antarmuka pengguna untuk sistem. Aktivitas terjadi selama fase Inception & Elaboration tetapi berlanjut melalui fase untuk meningkatkan desain yang sedang berlangsung. Pada fase konstruksi, *user stories* diimplementasikan & dikerjakan ulang secara iteratif untuk mencerminkan pemahaman tentang domain masalah saat proyek berlangsung
8	Dynamic System Development Method (DSDM)	Kebutuhan untuk proyek tertentu dikumpulkan & diperiksa untuk kelayakan & penentuan prioritas

Manajemen kebutuhan perangkat lunak merupakan bagian kritis dari proses pengembangan perangkat lunak. Dengan mengelola kebutuhan dengan baik, tim pengembangan dapat meningkatkan peluang kesuksesan proyek, mengurangi risiko kegagalan, dan memastikan bahwa perangkat lunak yang dihasilkan sesuai dengan harapan dan kebutuhan pengguna.

Manajemen kebutuhan, terlepas dari metodologi pengembangan perangkat lunak apa pun dilakukan untuk seumur hidup sistem. Terlihat di industri perangkat lunak saat mengembangkan produk perangkat lunak apa pun bahwa kebutuhan sering berubah dari sisi pelanggan yang menjadi sulit bagi pengembang perangkat lunak untuk menghasilkan perangkat lunak yang berkualitas. Tidak ada pendekatan yang tepat untuk mengelola persyaratan yang sering berubah selama siklus pengembangan perangkat lunak. Memenuhi kebutuhan pelanggan sangat penting. Metodologi pengembangan perangkat lunak *Agile* mendukung perubahan kebutuhan. Jadi, metodologi pengembangan perangkat lunak *Agile* yang berbeda dipelajari, untuk memberikan gambaran tentang bagaimana praktik manajemen kebutuhan dilakukan untuk pengembangan perangkat lunak. Manajemen kebutuhan memastikan bahwa kebutuhan pelanggan selalu didengar di semua titik dalam proses pengembangan dan pada saat yang sama menjaga integritas informasi tersebut untuk kehidupan sistem dan juga sehubungan dengan perubahan sistem dan lingkungannya.

DAFTAR PUSTAKA

Baruah, N. 2015. Requirement Management in Agile Software Environment. *The 2015 International Conference on Soft Computing and Software Engineering (SCSE)* (hal. 81-83). Procedia Computer Science. doi:https://doi.org/10.1016/j.procs.2015.08.414

Belfo, F. 2012. People, Organizational and Technological Dimensions of Software Requirement Specification. *CENTERIS 2012 - Conference on ENTERprise Information Systems / HCIST 2012 - International Conference on Health and Social Care Information Systems and Technologies* (hal. 310-318). Elsevier Ltd.

Burge, J. E. 2001. Knowledge Elicitation Tool Classification. Artificial Intelligence Research Group, Worcester Polytechnic Institute.

Davis, A. 1992. Operational prototyping: a new development approach. *IEEE Software*, 70-78.

Hofmann, H. F., & Lehner, F. 2001. Requirements Engineering as a Success Factor in Software Projects. *IEEE Software*.

Khan, S., Dulloo, A. B., & Verma, M. 2014. Systematic Review of Requirement Elicitation Techniques. *International Journal of Information and Computation Technology, 4*(2), 133-138.

Lamsweerde, A. v. 2000. Requirements engineering in the year 00. *Procreedings of the 22nd International Conference on Software Engineering - ICSE 00*, (hal. 5-19).

Lobo, L., & Arthur, J. D. 2005. Effective Requirements Generation: Synchronizing Early Verification & Validation, Methods and Method Selection Criteria. *ArXiv. /abs/cs/0503004*.

Object Management Group. 2011. Business Process Model and Notation (BPMN).

Object Management Group. 2017. Unified Modeling Language (UML). Diambil kembali dari https://www.omg.org/spec/UML/2.5.1/PDF

Pandey, D., Suman, U., & Ramani, A. K. 2010. An Effective Requirement Engineering Process Model for Software Development and Requirements Management. *2010 International Conference on Advances in Recent Technologies in Communication and Computing* (hal. 287-291). Kottayam, India: IEEE.

Ponsard, C., Massonet, P., Rifaut, A., & Molderez, J. F. 2005. Early verification and validation of mission critical systems. *Electronic Notes in Theoretical Computer Science*, 237-254.

Pressman, R. S. 2010. *Software Engineering: A Practitioner's Approach* (7th ed.). New York: McGraw-Hill.

Sommervile, I. 2011. *Software Engineering.* Boston: McGraw Hill.

Wiegers, K., & Beatty, J. 2013. *Software Requirements* (3rd ed.). Washington: Microsoft Press.