BAB II. TINJAUAN PUSTAKA

2.1. Studi Sebelumnya

Penelitian terkait *requirements engineering* telah banyak dilakukan dengan berbagai pendekatan, baik melalui *systematic literature review* maupun studi kasus. Beberapa penelitian berfokus pada pengembangan metode serta identifikasi tantangan dalam pengelolaan *requirement*. Selain itu, terdapat penelitian yang mengkaji penerapan *requirements engineering* dalam konteks implementasi sistem. Namun, penelitian yang secara khusus mengevaluasi kualitas *requirement documentation* dalam konteks proyek nyata masih terbatas.

Penelitian oleh Pujiharto dkk [15] mengkaji praktik dan tantangan *requirements engineering* pada konteks Industry 5.0 melalui pendekatan *systematic literature review*. Hasil penelitian menunjukkan bahwa pendekatan Agile banyak digunakan karena fleksibilitasnya dalam menghadapi perubahan kebutuhan, namun masih terdapat kendala seperti kompleksitas sistem, keamanan data, serta belum adanya standar yang konsisten. Kondisi ini menunjukkan bahwa meskipun fleksibilitas menjadi keunggulan, kebutuhan akan dokumentasi yang tetap terstruktur masih diperlukan untuk menjaga konsistensi *requirement*.

Penelitian Tomaselli dkk [16] membahas praktik *requirements management* dalam pengembangan perangkat lunak modern. Penelitian tersebut menunjukkan bahwa *requirement* perlu dikelola secara berkelanjutan karena adanya perubahan kebutuhan selama proses pengembangan. Selain itu, diperlukan mekanisme pengendalian dan pelacakan *requirement* untuk menjaga konsistensi sistem.

Penelitian oleh Alzayed [17] mengkaji praktik *requirements engineering* melalui pendekatan *literature review* dan studi kasus. Penelitian ini menyebutkan bahwa proses *requirements engineering* mencakup tahapan identifikasi, analisis, dokumentasi, validasi, serta pengelolaan *requirement.* Selain itu, penelitian tersebut menunjukkan bahwa kelemahan dalam proses *requirement*, seperti keterbatasan *elicitation*, kurangnya keterlibatan

stakeholder, serta proses *validation* dan *Verification* yang belum optimal, dapat berkontribusi terhadap ketidaksesuaian sistem yang dikembangkan.

Penelitian oleh Akbar dan Siregar [18] menggunakan pendekatan studi kasus pada institusi pengembangan perangkat lunak. Hasil penelitian menunjukkan bahwa tahap *elicitation* dan *requirement documentation* merupakan bagian penting dalam pengembangan sistem. Namun, masih ditemukan kondisi dimana kebutuhan tidak lengkap atau ambigu, serta dokumentasi yang belum terstruktur dengan baik. Hal tersebut berpotensi mempengaruhi kesesuaian antara kebutuhan dan sistem yang dihasilkan.

Penelitian oleh Qamar dkk [19] mengevaluasi kualitas dokumen *Software Requirements Specification* (SRS) melalui pendekatan eksperimen. Penelitian ini menggunakan aspek *completeness, consistency,* dan *ambiguity* sebagai indikator dalam menilai kualitas *requirement*. Hasil penelitian menunjukkan bahwa *requirement documentation* yang lebih terstruktur dapat membantu mengurangi ambiguitas serta meningkatkan kejelasan kebutuhan sistem.

Penelitian oleh Yoo et al. [20] berfokus pada pengembangan *traceability* antara *requirement* dan elemen arsitektur sistem. Hasil penelitian menunjukkan bahwa ketidaksesuaian antara kebutuhan dan implementasi sistem dapat terjadi ketika tidak terdapat hubungan yang jelas antara *requirement* dan desain sistem. Dengan adanya *traceability, requirement* dapat ditelusuri hingga tahap implementasi sehingga membantu menjaga kesesuaian antara kebutuhan dan sistem yang dikembangkan. Hal ini menunjukkan bahwa *requirement documentation* perlu didukung dengan keterlacakan agar dapat digunakan secara efektif dalam pengembangan sistem.

Penelitian oleh Mulholland dkk [21] membahas penerapan *Verification* dan *validation* (V&V) dalam lingkungan *model-based systems engineering*. Penelitian tersebut menjelaskan bahwa V&V merupakan proses penting untuk memastikan kesesuaian antara kebutuhan *stakeholder* dan sistem yang dikembangkan. Selain itu, proses V&V dilakukan sepanjang siklus hidup sistem dan mencakup berbagai bagian, mulai dari *requirement* hingga desain dan sistem. Hal ini menunjukkan bahwa proses *Verification* dan *validation* tidak

hanya dilakukan pada tahap akhir, tetapi menjadi bagian integral dalam pengembangan sistem secara keseluruhan.

Berdasarkan berbagai penelitian tersebut, dapat disimpulkan bahwa *Requirement documentation* memiliki peran penting dalam menjaga kesesuaian antara kebutuhan bisnis dan pengembangan sistem, terutama melalui pengelolaan kebutuhan, komunikasi antar *stakeholder*, serta penerapan *traceability* dan proses validasi. Namun, penelitian sebelumnya masih lebih banyak membahas metode dan tantangan dalam *requirements engineering,* sementara kajian yang secara khusus mengevaluasi penerapan *Requirement documentation* dalam organisasi untuk menilai kesesuaian antara kebutuhan bisnis dan hasil pengembangan sistem masih terbatas. Oleh karena itu, penelitian ini dilakukan untuk menganalisis penerapan *Requirement documentation* pada PT XYZ dalam konteks kesesuaian antara kebutuhan bisnis dan pengembangan sistem. Studi sebelumnya pada penelitian ini dapat dilihat pada Tabel 2.1.1.

Tabel 2.1.1. Studi Sebelumnya

No	Peneliti	Tahun	Metode	Tujuan	Temuan
1	Pujiharto dkk.	2024	*Systematic Literature Review* (SLR)	Mengidentifikasi praktik dan tantangan *requirements engineering* pada Industry 5.0	Agile menjadi metode dominan karena fleksibel, namun masih terdapat kendala seperti keamanan data, privasi, kompleksitas sistem, dan belum adanya standar yang jelas dalam proses *requirement*
2	Tomaselli	2025	Literatur + analisis praktik	Menganalisis praktik *requirements management* dalam pengembangan perangkat lunak modern	*Requirement* perlu dikelola secara berkelanjutan karena perubahan kebutuhan yang terjadi selama proses pengembangan. Penelitian ini juga menekankan pentingnya mekanisme pengendalian, pelacakan (*traceability*), serta pengelolaan perubahan untuk menjaga konsistensi antara *requirement* dan sistem. Tanpa pengelolaan yang baik, perubahan *requirement* dapat

No	Peneliti	Tahun	Metode	Tujuan	Temuan
					menyebabkan ketidaksesuaian dalam implementasi sistem.
3	Alzayed	2024	*Systematic Literature Review + Case Study*	Mengevaluasi praktik *requirements engineering* dalam proyek *E-Government*	Proses *requirements engineering* mencakup tahapan identifikasi, analisis, dokumentasi, validasi, dan pengelolaan *requirement*. Penelitian ini menunjukkan bahwa kelemahan dalam proses tersebut, seperti keterbatasan *elicitation*, kurangnya keterlibatan *stakeholder*, serta proses *validation* dan *Verification* yang belum optimal, dapat berkontribusi terhadap ketidaksesuaian antara kebutuhan dan sistem yang dikembangkan
4	Akbar & Siregar	2024	*Case Study*	Mengkaji praktik *requirements engineering* dalam institusi pengembangan perangkat lunak	Tahap *elicitation* dan *requirement documentation* merupakan bagian penting dalam pengembangan sistem. Namun, dalam praktiknya masih ditemukan *requirement* yang tidak lengkap, ambigu,

No	Peneliti	Tahun	Metode	Tujuan	Temuan
					serta dokumentasi yang belum terstruktur dengan baik. Kondisi ini berpotensi menyebabkan kesalahan interpretasi dan berdampak pada ketidaksesuaian antara kebutuhan awal dan sistem yang dihasilkan.
5	Qamar dkk	2025	Eksperimen	Mengevaluasi kualitas dokumen SRS	Kualitas *requirement* dapat diukur melalui aspek *completeness, consistency*, dan *ambiguity*. Penelitian ini menunjukkan bahwa *requirement documentation* yang lebih terstruktur dapat mengurangi ambiguitas, meningkatkan kejelasan kebutuhan, serta mempermudah pemahaman antar *stakeholder*. Hal ini berdampak pada peningkatan kualitas implementasi sistem.

No	Peneliti	Tahun	Metode	Tujuan	Temuan
6	Yoo et al.	2024	Studi konseptual / pengembangan model	Membangun keterlacakan (*traceability*) antara *requirement* dan arsitektur sistem	Ketidaksesuaian antara *requirement* dan implementasi sistem dapat terjadi ketika tidak terdapat hubungan yang jelas antara *requirement* dan desain sistem. Dengan adanya *traceability, requirement* dapat ditelusuri hingga tahap implementasi, sehingga membantu menjaga konsistensi, mempermudah analisis perubahan, serta meningkatkan kesesuaian antara kebutuhan dan sistem yang dikembangkan.
7	Mulholland dkk	2024	Studi konseptual / implementasi (MBSE)	Menganalisis penerapan *Verification & validation* (V&V) dalam sistem *engineering*	*Verification* dan *validation* (V&V) merupakan proses penting untuk memastikan bahwa sistem yang dikembangkan sesuai dengan *requirement* dan kebutuhan *stakeholder*. Proses ini dilakukan sepanjang siklus hidup sistem dan mencakup berbagai level, mulai dari

No	Peneliti	Tahun	Metode	Tujuan	Temuan
					requirement hingga desain dan sistem. V&V membantu mengidentifikasi ketidaksesuaian sejak awal serta memastikan bahwa hasil implementasi memenuhi kebutuhan yang telah ditentukan.

2.2. Dasar Teori

2.2.1. Sistem Informasi

Sistem informasi merupakan suatu sistem yang berfungsi untuk mengumpulkan, mengolah, menganalisis, dan mendistribusikan informasi guna mencapai tujuan tertentu. Sistem ini melibatkan berbagai komponen, seperti manusia, perangkat keras, perangkat lunak, serta basis data yang saling terintegrasi [22]. Keberadaan sistem informasi dapat mendukung proses komunikasi, memfasilitasi transaksi penting, serta membantu aktivitas manajerial dalam organisasi. Selain itu, sistem informasi juga berperan dalam menyediakan informasi yang relevan sehingga dapat mendukung pengambilan keputusan yang lebih tepat baik bagi individu maupun organisasi [23].

2.2.2. Pengembangan Sistem Informasi

Pengembangan sistem informasi merupakan proses terstruktur yang melibatkan tahapan mulai dari analisis kebutuhan hingga implementasi sistem. Proses ini bertujuan menghasilkan sistem yang sesuai dengan kebutuhan pengguna serta mendukung proses bisnis organisasi [22]. Salah satu tahap yang paling krusial dalam pengembangan sistem adalah pengumpulan dan pendefinisian kebutuhan, karena kebutuhan tersebut menjadi dasar bagi seluruh proses pengembangan. Ketidaktepatan dalam mendefinisikan kebutuhan dapat menyebabkan sistem yang dihasilkan tidak sesuai dengan tujuan bisnis yang diharapkan [1].

2.2.3. *Requirements Engineering*

Requirements engineering merupakan proses sistematis yang digunakan untuk mengidentifikasi, menganalisis, mendokumentasikan, serta memvalidasi kebutuhan sistem [24]. Proses ini menjadi bagian fundamental dalam pengembangan perangkat lunak karena berperan dalam menerjemahkan kebutuhan pengguna ke dalam spesifikasi sistem yang dapat diimplementasikan [1]. Selain itu, keberadaan *requirement* yang terdokumentasi dengan baik juga berkontribusi terhadap kualitas komunikasi dan keberhasilan pengembangan sistem [25].

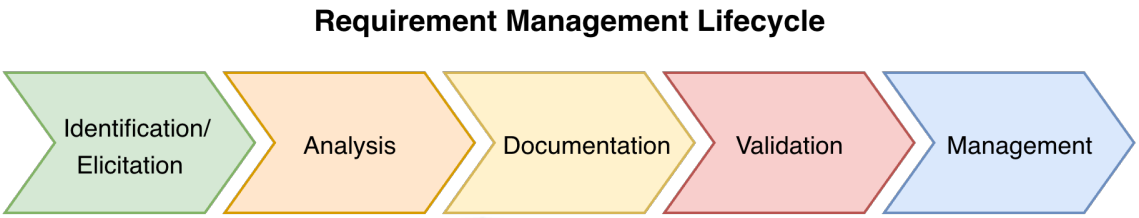
Dalam praktiknya, *requirements engineering* berperan penting dalam menjembatani komunikasi antara *stakeholder* dan tim pengembang sehingga kebutuhan sistem dapat dipahami dengan lebih baik. Meskipun pendekatan Agile mampu membantu mengurangi kesenjangan komunikasi dalam proses *requirements engineering*, penerapannya juga menimbulkan tantangan baru, seperti memastikan seluruh anggota tim memiliki pemahaman dan kompetensi yang memadai terhadap kebutuhan yang dikembangkan [24]. *Requirement* yang tidak terdefinisi dengan baik dapat menyebabkan ketidaksesuaian antara sistem yang dikembangkan dengan kebutuhan bisnis yang diharapkan [2]. Oleh karena itu, kualitas pada tahap ini sangat berpengaruh terhadap hasil akhir sistem yang dikembangkan.

Selain itu, *requirements engineering* menjadi dasar dalam penyusunan *requirement documentation* yang digunakan sepanjang siklus hidup sistem. Dalam proses ini juga terdapat aktivitas *Verification* dan *validation* (V&V) yang bertujuan untuk memastikan bahwa *requirement* telah ditulis secara jelas dan sesuai dengan kebutuhan pengguna. *Verification* berfokus pada pemeriksaan konsistensi dan kejelasan *requirement*, sedangkan *validation* memastikan kesesuaian *requirement* dengan kebutuhan bisnis. Kedua aktivitas ini mendukung kualitas *requirement* sebagai dasar pengembangan sistem [12].

2.2.4. *Requirement Management Lifecycle*

Requirements engineering dilakukan melalui beberapa tahapan yang saling berkaitan, yaitu identifikasi kebutuhan, analisis, dokumentasi, validasi, serta pengelolaan *requirement*. Tahapan ini dilakukan secara berkelanjutan sepanjang siklus pengembangan sistem. Penelitian menunjukkan bahwa *requirements engineering* mencakup aktivitas *identification, analysis, documentation, validation,* serta *management* yang dilakukan untuk memastikan *requirement* tetap relevan dan dapat diimplementasikan dengan baik [17]. Dalam praktiknya, *lifecycle* bersifat iteratif sehingga *requirement* dapat mengalami perubahan sesuai kebutuhan sistem. Dalam penelitian ini, *lifecycle* digunakan sebagai acuan dalam proses analisis, khususnya pada tahap

pengumpulan data, analisis dokumentasi, validasi, serta identifikasi gap dan rekomendasi.



Gambar 2.2.1. *Requirement Management Lifecycle* [17]

Gambar 2.2.1 menunjukkan tahapan *requirement management lifecycle* yang digunakan sebagai acuan dalam penelitian. Tahapan ini mencakup proses *elicitation, analysis, validation*, serta *management* terhadap *requirement*. Proses tersebut bersifat iteratif dan digunakan untuk mengevaluasi kesesuaian antara *requirement* dan sistem yang dikembangkan.

2.2.5. *Requirement documentation*

Requirements Documentation merupakan dokumentasi kebutuhan sistem yang digunakan untuk mengomunikasikan informasi kebutuhan kepada tim pengembang maupun *stakeholder* terkait [26]. Melalui dokumentasi yang terstruktur, kebutuhan sistem dapat dipahami, ditinjau, dan digunakan sebagai acuan bersama dalam proses pengembangan sistem. *Requirement documentations* menjadi media komunikasi utama antara *stakeholder*, analis, dan pengembang dalam memahami sistem yang akan dibangun [27]. Dengan adanya dokumentasi yang baik, kebutuhan sistem dapat disampaikan secara konsisten dan mengurangi perbedaan pemahaman [25].

Dokumentasi *requirement* diperlukan sebagai acuan seluruh aktivitas pengembangan sistem sehingga setiap *stakeholder* memiliki pemahaman yang sama terhadap kebutuhan yang akan diimplementasikan [28]. Dokumen kebutuhan mencakup kebutuhan fungsional dan non-fungsional, ruang lingkup sistem, serta kebutuhan *stakeholder*. Standar seperti IEEE 830-1998 maupun ISO/IEC/IEEE 29148 memberikan panduan dalam penyusunan *Requirement documentations* agar sistem yang dikembangkan memiliki kualitas yang baik

[11], [12]. Oleh karena itu, dokumentasi harus disusun secara sistematis, jelas, dan mudah dipahami agar dapat digunakan sepanjang siklus hidup sistem [29].

Selain itu, dokumentasi kebutuhan juga mendukung proses *Verification* dan *validation* kebutuhan [30]. Dokumentasi yang jelas memungkinkan *requirement* untuk ditinjau, dianalisis, serta diperiksa konsistensinya secara lebih terstruktur. Dengan demikian, dokumentasi kebutuhan berperan dalam menjaga kualitas dan konsistensi kebutuhan selama proses pengembangan sistem [1].

Requirement documentations juga berperan dalam mendukung proses *traceability* dan pengelolaan perubahan kebutuhan. Dengan adanya dokumentasi yang jelas, setiap perubahan *requirement* dapat dilacak dan dianalisis dampaknya terhadap sistem yang dikembangkan. Hal ini membantu pengembang dalam menjaga konsistensi antara *requirement*, desain, dan implementasi sistem. Oleh karena itu, *Requirement documentations* menjadi elemen penting dalam menjaga kualitas dan keberlanjutan sistem informasi [31].

2.2.6. Pengalaman *Stakeholder* Pengguna *Requirement Documentation*

Pengalaman *stakeholder* pengguna *requirement documentation* dalam penelitian ini merujuk pada pengalaman *stakeholder* ketika menggunakan dokumen *requirement* sebagai acuan dalam proses pengembangan sistem. Pengguna yang dimaksud bukan pengguna akhir sistem informasi, melainkan pihak yang terlibat langsung dalam penggunaan dokumen *requirement*, seperti *Business Development Strategist*, *Business Requirement Analyst*, *Developer*, dan *Quality Assurance*. Dalam konteks pengembangan perangkat lunak, *requirement documentation* berperan untuk memperjelas kebutuhan, mendukung komunikasi antar *stakeholder*, serta membantu proses pengembangan agar *requirement* dapat dipahami secara konsisten [26]. Dokumentasi *requirement* yang baik juga membantu mengurangi kesenjangan komunikasi dan mendukung pemahaman bersama terhadap kebutuhan sistem [19].

Dalam penelitian ini, pengalaman *stakeholder* dianalisis berdasarkan bagaimana *stakeholder* memahami, menggunakan, memvalidasi, dan memanfaatkan *requirement documentation* selama proses pengembangan sistem informasi di PT XYZ. Pengalaman tersebut mencakup penggunaan

dokumen dalam mendukung *requirement engineering*, *development*, *testing*, komunikasi antar *stakeholder*, pengelolaan perubahan *requirement*, serta kesesuaian antara kebutuhan bisnis dan sistem yang dikembangkan. Selain itu, pengalaman *stakeholder* juga dilihat dari penilaian terhadap kualitas dokumen, khususnya aspek *completeness*, *clarity*, dan *consistency*, karena ketiga aspek tersebut memengaruhi kemudahan *stakeholder* dalam memahami dan menggunakan *requirement* [19]. Oleh karena itu, analisis pengalaman *stakeholder* diperlukan untuk mengetahui sejauh mana *requirement documentation* dapat mendukung proses pengembangan sistem secara lebih terstruktur [26].

2.2.7. *Requirement Management*

Requirements management merupakan proses pengelolaan kebutuhan sistem sepanjang siklus hidup pengembangan. Proses ini mencakup aktivitas seperti pengendalian perubahan, pelacakan *requirement*, serta memastikan bahwa setiap *requirement* tetap relevan dengan tujuan bisnis. Dalam praktiknya, perubahan *requirement* sering terjadi seiring dengan perkembangan kebutuhan organisasi. Oleh karena itu, diperlukan mekanisme yang jelas untuk mengelola perubahan tersebut agar tetap terkendali [16].

Pengelolaan *requirement* sangat bergantung pada dokumentasi kebutuhan yang jelas dan terstruktur. Dokumentasi memungkinkan setiap kebutuhan dicatat, dipahami, serta digunakan sebagai acuan dalam mengelola perubahan. Tanpa dokumentasi yang memadai, perubahan *requirement* berpotensi tidak terkontrol dan dapat menyebabkan ketidaksesuaian antara sistem yang dikembangkan dengan kebutuhan pengguna. Oleh karena itu, *Requirement documentation* menjadi dasar dalam pelaksanaan *requirements management* .

Selain itu, perubahan *requirement* yang tidak terkelola dengan baik dapat berdampak pada peningkatan kompleksitas pengembangan sistem. Kondisi ini dapat memicu risiko seperti keterlambatan proyek serta kesulitan dalam menjaga konsistensi sistem. Dengan adanya *requirements management* yang terstruktur, setiap perubahan dapat dianalisis dan dikendalikan sehingga proses

pengembangan sistem dapat berjalan lebih terarah dan sesuai dengan kebutuhan yang telah ditetapkan [9].

2.2.8. *Requirement Traceability*

Requirements traceability adalah kemampuan untuk melacak hubungan antara *requirement* dengan artefak lain dalam pengembangan sistem, seperti desain, implementasi, dan pengujian. *Traceability* membantu dalam memahami asal-usul *requirement* serta dampak dari setiap perubahan yang terjadi. Dengan adanya *traceability*, proses pengembangan menjadi lebih terstruktur dan transparan. Hal ini juga memudahkan dalam proses validasi dan verifikasi sistem [31].

Keterlacakan berperan dalam memastikan bahwa seluruh kebutuhan telah diimplementasikan sesuai dengan kebutuhan yang telah ditentukan melalui keterkaitan yang jelas antara kebutuhan dan implementasi sistem. Dengan adanya hubungan tersebut, risiko kesalahan dapat diminimalkan serta mendukung proses evaluasi sistem dan pengelolaan perubahan kebutuhan secara lebih terstruktur. Namun, *traceability* tidak dapat dilakukan secara efektif tanpa adanya *Requirement documentation* yang tersusun dengan baik sebagai dasar pelacakan. Dokumentasi yang jelas memungkinkan setiap *requirement* memiliki keterkaitan yang dapat ditelusuri dengan elemen sistem lainnya, sehingga setiap perubahan dapat dianalisis secara lebih terarah. Oleh karena itu, hubungan antara *Requirement documentation* dan *traceability* menjadi penting dalam menjaga konsistensi dan kualitas pengembangan sistem informasi..

2.2.9. Kesesuaian Kebutuhan Bisnis dan Sistem Informasi

Kesesuaian antara kebutuhan bisnis dan sistem informasi merupakan aspek penting dalam memastikan bahwa sistem yang dikembangkan dapat mendukung kebutuhan organisasi secara efektif. Sistem informasi tidak hanya berfungsi sebagai alat operasional, tetapi juga harus mampu merepresentasikan dan mendukung kebutuhan bisnis. Oleh karena itu, sistem yang dikembangkan perlu selaras dengan tujuan bisnis serta aktivitas operasional yang ada dalam organisasi. Keselarasan tersebut penting untuk memastikan bahwa sistem yang

dikembangkan mampu memenuhi kebutuhan *stakeholder* dan memberikan nilai bagi organisasi, terutama dalam lingkungan bisnis yang dinamis dan terus mengalami perubahan [32]. Ketidaksesuaian antara sistem dan proses bisnis dapat menyebabkan sistem tidak dapat digunakan secara optimal [33].

Konsep kesesuaian ini sering dikaitkan dengan *alignment antara business strategy, business process,* dan sistem informasi. *Alignment* terjadi ketika kebutuhan bisnis, proses yang dijalankan, serta sistem yang dikembangkan berada dalam kondisi yang saling mendukung. Pendekatan *requirements engineering* berperan dalam menghubungkan kebutuhan bisnis ke dalam spesifikasi sistem sehingga kesesuaian tersebut dapat dicapai. Dengan adanya keterkaitan antara *requirement* dan proses bisnis, sistem dapat dikembangkan secara lebih terarah dan sesuai dengan tujuan organisasi [33].

Requirement documentations menjadi media yang menghubungkan kebutuhan bisnis dengan sistem yang dikembangkan. Melalui dokumentasi yang terstruktur, kebutuhan bisnis dapat diterjemahkan secara lebih jelas ke dalam spesifikasi sistem sehingga mendukung tercapainya *alignment* antara kebutuhan bisnis dan sistem informasi. Selain itu, kesesuaian antara kebutuhan bisnis dan sistem informasi juga dipengaruhi oleh kualitas *requirement* yang dihasilkan. *Requirement* yang tidak lengkap, tidak jelas, atau tidak konsisten dapat berdampak negatif terhadap proses pengembangan sistem dan berpotensi menyebabkan ketidaksesuaian antara kebutuhan dan hasil implementasi [4]. Hal ini menunjukkan bahwa dokumentasi kebutuhan memiliki peran dalam menjaga keselarasan antara kebutuhan bisnis dan sistem yang dikembangkan. Dengan demikian, kesesuaian tersebut dapat dicapai melalui proses *requirements engineering* dan dokumentasi yang terstruktur.

2.2.10. *Scope Creep* dalam Pengembangan Sistem

Scope creep merupakan kondisi di mana terjadi perubahan atau penambahan ruang lingkup proyek secara tidak terkontrol selama proses pengembangan sistem. Perubahan ini umumnya muncul akibat *requirement* yang tidak didefinisikan secara jelas atau tidak terdokumentasi dengan baik sejak awal. *Scope creep* dapat menyebabkan meningkatnya kompleksitas proyek serta

ketidakjelasan arah pengembangan sistem. Oleh karena itu, kejelasan *requirement* menjadi faktor penting dalam menjaga stabilitas ruang lingkup proyek [9].

Dalam pengembangan sistem informasi, *scope creep* sering berkaitan dengan proses *requirements engineering*, khususnya pada tahap pengumpulan dan analisis kebutuhan. *Requirement* yang ambigu atau tidak lengkap dapat memicu perubahan berulang selama proses pengembangan. Selain itu, kurangnya pengelolaan perubahan *requirement* juga dapat memperbesar risiko terjadinya *scope creep.* Kondisi ini menunjukkan bahwa kualitas *requirement* sangat berpengaruh terhadap keberlangsungan proyek [34].

Salah satu dampak dari pengelolaan *requirement* yang tidak optimal adalah terjadinya *scope creep*, yaitu perubahan ruang lingkup proyek yang tidak terkontrol, yang dapat memengaruhi biaya, waktu, dan kualitas hasil pengembangan [35]. Oleh karena itu, diperlukan *Requirement documentation* yang jelas serta pengelolaan perubahan yang terstruktur untuk mengendalikan ruang lingkup proyek. Dengan adanya dokumentasi dan *requirements management* yang baik, perubahan yang terjadi dapat dikontrol sehingga risiko *scope creep* dapat diminimalkan [9].

2.2.11. Peran Dokumentasi Kebutuhan terhadap Keberhasilan Proyek

Requirement documentation merupakan salah satu faktor keberhasilan proyek [23]. Dokumentasi yang disusun secara jelas dan terstruktur membantu memastikan bahwa kebutuhan pengguna dapat dipahami secara konsisten oleh seluruh *stakeholder* yang terlibat. Dengan adanya pemahaman yang sama mengenai kebutuhan sistem, risiko kesalahan interpretasi dapat dikurangi sehingga proses pengembangan dapat berjalan lebih terarah dan sesuai dengan tujuan proyek [25],[23].

Dokumentasi kebutuhan juga berperan dalam mendukung koordinasi dan komunikasi antar *stakeholder*, analis, pengembang, serta tim pengujian. Kebutuhan yang terdokumentasi dengan baik memudahkan proses pengambilan keputusan, verifikasi, dan validasi *requirement* selama pengembangan sistem berlangsung. Kondisi tersebut membantu mengurangi potensi *rework* akibat

kesalahan pemahaman kebutuhan serta meningkatkan kualitas sistem yang dihasilkan [25].

Selain itu, dokumentasi kebutuhan mendukung pengelolaan perubahan kebutuhan secara lebih terkontrol. Setiap perubahan dapat ditelusuri, dianalisis dampaknya, dan dikomunikasikan kepada pihak yang terlibat sebelum diimplementasikan ke dalam sistem. Kemampuan dalam mengelola perubahan ini membantu menjaga ruang lingkup proyek, mengurangi risiko keterlambatan maupun pembengkakan biaya, serta memastikan sistem yang dikembangkan tetap sesuai dengan kebutuhan bisnis organisasi [7]. Oleh karena itu, *Requirement documentation* tidak hanya berfungsi sebagai sarana pencatatan kebutuhan, tetapi juga menjadi salah satu faktor yang mendukung keberhasilan proyek melalui peningkatan kualitas komunikasi, pengendalian perubahan, efisiensi proses pengembangan, dan kesesuaian hasil sistem dengan kebutuhan *stakeholder*. [25].

2.2.12. *Directed Qualitative Content Analysis* (DQCA)

Content Analysis merupakan metode yang digunakan untuk menganalisis isi komunikasi atau data tekstual secara sistematis. Data yang dianalisis dapat berupa transkrip wawancara, dokumen, pernyataan tertulis, laporan, maupun bentuk komunikasi lainnya. Metode ini bertujuan mengelompokkan isi data ke dalam kategori tertentu agar peneliti dapat memperoleh pemahaman terhadap fenomena yang diteliti. *Content Analysis* dapat digunakan melalui pendekatan kuantitatif maupun kualitatif, bergantung pada tujuan dan cara data dianalisis [36].

Qualitative Content Analysis berfokus pada interpretasi makna yang terdapat dalam data tekstual melalui proses pengkodean dan klasifikasi yang sistematis. Pendekatan ini tidak hanya memperhatikan kemunculan kata atau frekuensi kategori, tetapi juga mempertimbangkan konteks, maksud, dan hubungan antarbagian data. Hasil analisis dapat berupa kategori, pola, dan pemahaman yang menjelaskan fenomena berdasarkan data yang dikumpulkan. *Qualitative Content Analysis* dapat dilakukan secara induktif maupun deduktif sesuai dengan sumber pembentukan kategori analisis [37].

Pada pendekatan induktif, kategori dikembangkan secara langsung dari data, sedangkan pada pendekatan deduktif kategori awal disusun berdasarkan teori, kerangka konseptual, atau penelitian terdahulu. Pendekatan deduktif dalam Qualitative Content Analysis disebut sebagai Directed Qualitative Content Analysis. Pendekatan ini digunakan ketika penelitian telah memiliki dasar teori atau kerangka analisis yang dapat digunakan untuk mengarahkan proses pengkodean data. Data empiris kemudian digunakan untuk mendukung, memperluas, menyempurnakan, atau menunjukkan variasi terhadap kategori awal tersebut [37], [38].

Meskipun menggunakan kategori yang telah ditentukan sebelumnya, Directed Qualitative Content Analysis tetap memungkinkan munculnya kode, subkategori, atau kategori tambahan. Data yang tidak dapat dimasukkan secara tepat ke dalam kategori awal perlu dianalisis kembali agar tidak diabaikan atau dipaksakan ke dalam kerangka yang tidak sesuai. Dengan demikian, teori berfungsi sebagai kerangka awal analisis, sedangkan data empiris tetap menjadi dasar dalam penyusunan interpretasi dan temuan penelitian. Pendekatan ini membantu menjaga keterkaitan antara teori, data, dan fokus penelitian secara sistematis [37], [38].

Dalam penelitian ini, Directed Qualitative Content Analysis digunakan karena kategori analisis telah disusun berdasarkan pertanyaan penelitian, tujuan penelitian, teori Requirements Engineering, dan penelitian terdahulu. Kategori tersebut digunakan untuk menganalisis hasil wawancara dan studi dokumentasi mengenai penerapan *requirement documentation* dalam pengembangan sistem informasi di PT XYZ. Penggunaan pendekatan ini memungkinkan data dari *stakeholder* dan dokumen dianalisis menggunakan kerangka yang konsisten tanpa menghilangkan informasi tambahan yang ditemukan selama proses analisis. Penjelasan mengenai tahapan penerapannya dalam penelitian disajikan lebih lanjut pada Bab III.

2.2.13. Keabsahan Data penelitian kualitatif

Keabsahan data dalam penelitian kualitatif digunakan untuk memastikan bahwa hasil penelitian dapat dipercaya dan sesuai dengan kondisi yang diteliti.

Keabsahan data berkaitan dengan konsep *trustworthiness*, yang mencakup kredibilitas, dependabilitas, konfirmabilitas, dan transferabilitas hasil penelitian [39]. Penerapan keabsahan data penting dilakukan karena penelitian kualitatif melibatkan proses interpretasi peneliti, sehingga diperlukan mekanisme untuk memastikan bahwa temuan yang dihasilkan tetap didasarkan pada data yang diperoleh.

Dalam penelitian ini, keabsahan data dilakukan melalui triangulasi sumber, triangulasi metode, dan *peer debriefing*. Triangulasi sumber dilakukan dengan membandingkan informasi dari narasumber yang memiliki peran berbeda, sedangkan triangulasi metode dilakukan dengan membandingkan hasil wawancara dan studi dokumentasi [40]. Sementara itu, *peer debriefing* dilakukan dengan meminta pihak yang memahami konteks penelitian untuk meninjau kategori analisis dan hasil interpretasi agar temuan yang dihasilkan tetap sesuai dengan data dan konteks penggunaan *requirement documentation* di PT XYZ [40].