

PEDOMAN, TANTANGAN, DAN MANFAAT IMPLEMENTASI DEVOPS UNTUK PENGEMBANGAN DAN OPERASI PERANGKAT LUNAK: TINJAUAN LITERATUR SISTEMATIS

Meinahen B. Bansoma

Program Studi Teknologi Informasi Universitas Citra Bangsa
Jl. Manafe No.17 Kel. Kayu Putih Kec. Oebobo Kupang-NTT

Abstrak

DevOps adalah pengembangan perangkat lunak yang menyediakan respon tercepat. Oleh karena itu, dalam mengembangkan perangkat lunak elemen pengetahuan yang berhubungan dengan pedoman, tantangan, serta manfaat dari implementasi DevOps harus dipahami. Namun, dalam praktiknya yang sangat kompleks masih banyak SDM pada organisasi yang belum memahami tentang pemanfaatan DevOps. Oleh karena itu, pada studi ini akan dilakukan tinjauan secara sistematis mengenai pedoman, tantangan, serta manfaat dalam mengimplementasikan DevOps untuk pengembangan dan operasi perangkat lunak. Studi dimulai dengan melakukan proses seleksi literatur menggunakan beberapa kriteria seperti nama jurnal, tahun terbit, negara, jenis penelitian, alat ekstraksi data serta banyak author. Literatur yang telah diseleksi kemudian akan direview menggunakan kriteria inklusif dan analisa ketidaksesuaian pada literatur. Hasilnya kami berhasil melakukan seleksi terhadap 35 literatur dengan kriteria sesuai dan cukup sesuai dengan masalah yang akan diidentifikasi secara luas. Berdasarkan 35 literatur tersebut, maka dapat diidentifikasi tiga poin utama yaitu pedoman, manfaat, dan tantangan dalam mengimplementasikan DevOps. Berdasarkan tinjauan yang telah dilakukan, maka ditemukan bahwa pipeline CI/CD dapat diterapkan untuk meningkatkan kualitas, membangun, menguji dan rilis dengan cepat dan andal serta mengembangkan perangkat lunak modern. Studi ini diharapkan dapat berkontribusi pada (1) Koseptualisasi implementasi DevOps dalam mengembangkan serta operasi perangkat lunak, (2) Tinjauan singkat mengenai pedoman, tantang serta manfaat dari DevOps, (3) pemahaman baru kepada SDM pada organisasi dalam mengimplementasikan DevOps..

Kata Kunci - Pedoman, Manfaat, Tantangan implementasi DevOps, pipeline CI/CD

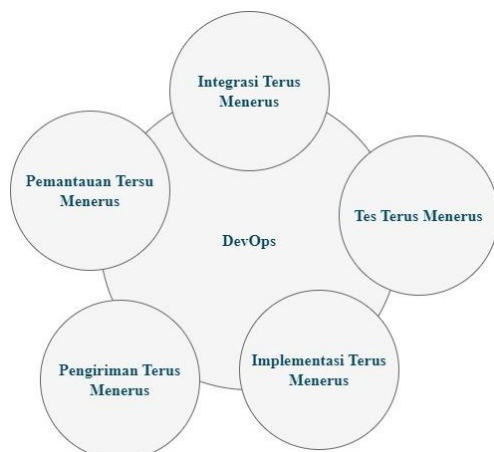
I. PENDAHULUAN

Dalam melakukan pengembangan perangkat lunak banyak organisasi berusaha untuk mengimplementasikan praktek pengembangan dan operasi perangkat lunak, salah satu pendekatan yang sekarang banyak digunakan adalah DevOps [4]. Istilah DevOps mulai digunakan pada tahun 2009 di belgia saat diselenggarakan konferensi migrasi pusat data, dengan harapan departemen pengembangan dapat bertanggungjawab pada penerapan fitur-fitur baru yang diperlukan untuk memenuhi kebutuhan bisnis sedangkan departemen operasi (Ops) bertanggungjawab dalam memodifikasi pada tingkat produksi dan layanan. Gambar 1 menunjukkan 6 karakteristik paling penting dari DevOps [8].



Gambar 1. 1 Fitur Devops [8]

DevOps terus mengalami perkembangan, ditahun 2014 DevOps mengalami ekspansi dengan ditingkatkan ke dalam lingkup perusahaan. Hal ini ditunjukkan dengan diluncurkannya DevOps Enterprise Summit. Fenomena DevOps semakin dikenal berdasarkan kemampuannya dalam mendukung pengiriman nilai secara berkelanjutan dan dapat melakukan perubahan [10].



Gambar 1.2 Proses DevOps [8, 9]

DevOps adalah seperangkat praktek yang menggabungkan pengembangan perangkat lunak (Dev) dan operasi TI (Ops) untuk menghasilkan perangkat lunak berkualitas tinggi dengan cara yang lebih efisien dan lebih cepat [6]. DevOps merupakan pengembangan dari pendekatan Agile dengan tujuan dapat menciptakan kontinuitas sistem alur kerja dari pengembangan hingga operasi dapat berjalan lebih lancar tanpa adanya hambatan menggunakan *Continuous Integration (CI)*, *Continuos Deployment (CD)*, *Continuous Delivery (CDE)*, serta *Continuous Feedback* [9, 10]. Dalam Praktek pengembangan perangkat lunak DevOps telah banyak berkontribusi dalam mengembangkan dan mengoperasionalkan sistem perangkat lunak, serta pada beberapa praktek penggunaan DevOps dapat memberikan pengetahuan yang diperlukan untuk operasionalisasi [7] Gambar 2 mengilustrasikan proses DevOps tentang bagaimana pentingnya kontinuitas merupakan elemen penting dalam mengembangkan perangkat lunak.

Pada penelitian [11] mengungkapkan bahwa DevOps merupakan metode yang telah banyak digunakan perusahaan dalam mengembangkan perangkat lunak berkualitas berdasarkan fitur produk melalui penerapan proses otomatisasi, sehingga dapat mempercepat siklus penerapan, menurunkan tingkat kegagalan, mempercepat waktu rilis, dan mempersingkat waktu dalam melakukan perbaikan. Hal serupa juga dikemukakan pada penelitian [12] dengan memanfaatkan DevOps, maka praktek penggunaan seperti pengontrolan versi, kontinuitas integrasi, pengujian, penerapan dan otomatisasi menjadi elemen penting dalam menyebarkan penyimpanan dan aplikasi basis data, hal ini sangat berguna karena dapat memastikan konsistensi, pengulangan, dan keandalan. Oleh karena itu, adopsi DevOps dalam mengembangkan perangkat lunak dapat memberikan keuntungan yang lebih efektif dan efisien dalam pengembangan serta operasi perangkat lunak, seperti yang telah ditunjukkan pada penelitian [6][8][9]. Meskipun adopsi DevOps dalam mengembangkan perangkat lunak menunjukkan banyak manfaat akan tetapi dalam mengimplementasikan akan menjadi tantangan pada pengembangan dan operasi perangkat lunak karena belum memiliki pedoman standar umum, kurang jelasnya manfaatnya, serta minimnya pengetahuan tentang tantangan pada saat akan menggunakan DevOps. Tantangan ini akan meningkat signifikan dan kompleks ketika implementasinya mempertimbangkan pengembangan perangkat lunak dengan kondisi dan kebiasaan yang sedang terjadi [7][32].

Beberapa Tinjauan Literatur Sitematis (SLR) telah dilakukan, seperti pada [9] studi ini bertujuan untuk meninjau dan menganalisis secara sistematis tantangan dan praktik yang dihadapi serta diadopsi dalam kontinuitas integrasi untuk meningkatkan perangkat lunak di DevOps. Studi ini mengemukakan bahwa DevOps membangun kualitas ke dalam seluruh rantai pengiriman perangkat lunak dengan menekankan pada komunikasi, kolaborasi, dan integrasi di antara berbagai stakeholder dalam proses pengembangan perangkat lunak, yaitu pengembangan, QA, dan operasi. Akan tetapi DevOps dalam mengembangkan perangkat lunak juga memiliki tantangannya tersendiri. Tantangan-tantangan ini diidentifikasi yang kemudian dikategorikan dalam berbagai kapasitas sebagai teknis. Hasilnya ditemukan 26 tantangan dan 28 praktik yang sebagian besar dalam integrasi kontinuitas, otomatisasi, alat, pemantauan, dan *pipeline*. Praktik-praktik ini memiliki bukti peningkatan kualitas melalui rilis yang lebih cepat, pemantauan kinerja, pengurangan risiko, pengurangan waktu dan upaya pengujian, keamanan yang lebih baik, putaran umpan balik yang cepat. Sedangkan studi [17] dilakukan dengan tujuan untuk memberi pembaca pemahaman mendalam tentang adopsi DevOps, termasuk mengantisipasi kemungkinan keuntungan dan kerugiannya bagi organisasi untuk meningkatkan proses pengembangan perangkat lunak mereka. Studi ini juga mengemukakan bahwa organisasi yang telah mengadopsi DevOps telah mendapatkan peningkatan yang signifikan dalam frekuensi rilis perangkat lunak, peningkatan rata-rata 55% dalam waktu ke pasar, dan pengurangan kesalahan penerapan sebesar lima puluh poin persentase. Namun, adopsi DevOps juga menimbulkan tantangan bagi organisasi, seperti resistensi budaya, kompleksitas teknis, dan masalah keamanan..

II. Dasar Teori

Singh, A [19] melakukan studi dengan memperhatikan peningkatan minat pada literatur tentang praktik kontinuitas dan integrasi, maka penting untuk dilakukan peninjauan secara sistematis dan menggabungkan pendekatan, alat, tantangan, dan praktik dalam mengadopsi dan menerapkan praktik kontinuitas. Studi ini mengemukakan bahwa CI/CD merupakan bagian konsep dari DevOps serta memiliki tujuan untuk mengirimkan perubahan perangkat lunak dengan cepat ke pengguna akhir melalui pembuatan, pengujian, dan penerapan otomatis. Di sisi lain, DevOps muncul sebagai metodologi yang terlibat dalam seluruh siklus hidup produk melalui pemasaran, perencanaan, sumber daya manusia, dan penjualan seiring dengan pengembangan dan operasi. Karena

meningkatnya persaingan dalam industri perangkat lunak, organisasi sebagai peran utama dalam menetapkan sumber daya yang diperlukan untuk mengembangkan dan memberikan produk yang dapat dipercaya dan berkualitas tinggi kepada konsumen. Sementara pendekatan, alat, dan praktik yang dilaporkan menangani berbagai tantangan, ada beberapa tantangan dan kesenjangan yang memerlukan penelitian di masa depan untuk meningkatkan pemahaman dan pelaporan informasi kontekstual dalam studi yang melaporkan berbagai aspek praktik berkelanjutan; mendapatkan pemahaman mendalam tentang bagaimana sistem intensif perangkat lunak harus dirancang (ulang) untuk mendukung praktik berkelanjutan; mengatasi kurangnya pengetahuan dan alat untuk proses rekayasa dalam merancang dan menjalankan jaringan pipa penerapan yang aman.

Hamza, U., dkk [32] melakukan tinjauan studi tentang manfaat, tantangan, serta panduan dalam mengimplementasi perangkat lunak dengan mengadopsi DevOps. Pada studi ini ditemukan bahwa otomatisasi, sharing, dan proses kolaborasi DevOps sangat berkaitan erat dengan manfaat dan keberhasilan dalam pengembangan perangkat lunak berdasarkan pedoman umum. Studi ini juga menemukan bahwa (1) implementasi pengembangan dan operasi perangkat lunak tanpa pedoman dan berbagai tantangan DevOps dapat mengakibatkan kurangnya pengetahuan yang berhubungan dengan proses otomatisasi, (2) tidak ada pendekatan ilmiah yang transparan dalam penentuan pemanfaatan DevOps yang berhubungan dengan proses, praktik, alat, serta implementasi, (3) DevOps adalah pengembangan perangkat lunak yang menyediakan respon tercepat. Oleh karena itu, dalam mengembangkan perangkat lunak elemen pengetahuan yang berhubungan dengan pedoman, tantangan, serta manfaat dari implementasi DevOps harus dipahami. Namun, dalam praktiknya yang sangat kompleks masih banyak SDM pada organisasi yang belum memahami tentang pemanfaatan DevOps.

Berdasarkan beberapa literatur yang telah dibahas, maka pada studi ini akan dilakukan tinjauan secara sistematis mengenai pedoman, tantangan, serta manfaat dalam mengimplementasikan DevOps untuk pengembangan dan operasi perangkat lunak. Studi ini akan difokuskan pada proses otomatisasi, Continuous

Integration (CI), Continuous Deployment (CD), Continuous Delivery (CDE), serta Continuous Feedback. Studi ini diharapkan dapat berkontribusi pada (1) Koseptualisasi implementasi DevOps dalam mengembangkan serta operasi perangkat lunak, (2) Tinjauan singkat mengenai pedoman, tantang serta manfaat dari DevOps, (3) pemahaman baru kepada SDM pada organisasi dalam mengimplementasikan DevOps [35].

III. METODE PENELITIAN

Tahapan pada studi ini diadopsi berdasarkan penelitian [31], [32]. Prosedur SLR dapat dimanfaatkan dalam meninjau pengembangan perangkat lunak dengan mengimplementasikan DevOps [32]. Tahapan peninjauan pada studi ini dilakukan berdasarkan tiga (3) tahap utama:

a. Pembuatan Rencana Tinjauan Literatur

Studi ini dilakukan dengan tujuan untuk memahami adopsi pedoman, tantangan, dan manfaat dari implementasi DevOps dalam pengembangan dan operasi perangkat lunak serta meninjau berbagai pendekatan yang diperlukan. Oleh karena itu, untuk menjawab tujuan studi ini, maka akan dibuat pertanyaan penelitian yang akan digunakan sebagai pedoman dalam meninjau literatur sebelumnya:

RQ1 apa pedoman implementasi DevOps dalam mengembangkan perangkat lunak?

RQ2 apa tantangan implementasi DevOps dalam mengembangkan perangkat lunak?

RQ3 apa manfaat implementasi DevOps dalam mengembangkan perangkat lunak?

b. Review Literatur

Tahapan awal dari tinjauan pada studi ini dengan mengidentifikasi studi primer yang relevan dan kompatibel. Pencarian literatur menggunakan 5 perpustakaan digital dengan predikat terbaik dalam menyajikan referensi yang relevan: SCOPUS, ScienceDirect, Google Scholar, SpingerLink, EBSCO, dan Wiley Online Library. Proses pencarian akan dijelaskan lebih lanjut pada Tabel 1.

S/No Kriteria Inklusi Kriteria Pengecualian

1 Literatur sebelumnya yang menyajikan konsep/defenisi dan pendekatan implementasi DevOps dalam mengembangkan perangkat lunak Literatur secara implisit memberikan solusi untuk Tantangan dan latihan DevOps

2 Literatur sebelumnya yang menyajikan strategi serta praktik implementasi DevOps Literatur menyajikan survey

3 Literatur sebelumnya yang menyajikan masalah/tujuan, model serta manfaat mengimplementasi DevOps Editorial, abstrak, panel tesis, monograf, buku, bagian buku

4 Literatur sebelumnya yang menyajikan prinsip-

prinsip implementasi DevOps Jurnal tidak ditulis dalam Bahasa Inggris

5 Literatur sebelumnya yang menyajikan studi kasus yang masih memiliki hubungan dengan pedoman implementasi DevOps Literatur yang dipublish diatas tahun 2019

Tabel 1. Kriteria Inklusi dan Pengecualian

Daftar pemeriksaan literatur (pada tabel 2) akan dimanfaatkan sebagai bahan acuan dalam menilai kualitas dari referensi literatur sebelumnya. Evaluasi pada setiap pernyataan akan dihitung berdasarkan skala numerik yang diklasifikasi kedalam tiga poin: sesuai (dua poin), cukup sesuai (1), tidak Sesuai (0). Dalam studi ini terdapat 5 pernyataan yang akan digunakan dalam mengevaluasi tingkat kesesuaian literatur dalam daftar pemeriksaan. Berdasarkan hasil tersebut maka akan ditetapkan jumlah poin maksimum pada literatur sebelumnya adalah 10 poin dengan minimal poin terendah adalah 0. Oleh karena itu, literatur sebelumnya harus mendapatkan 5 poin sehingga dapat digunakan sebagai referensi pada studi ini. Sedangkan literatur dengan kualifikasi poin 4 atau lebih rendah maka akan didiskualifikasi dari studi ini. Dalam studi ini penting untuk dipahami bahwa literatur yang akan dinilai akan dipilih berdasarkan relevansi yang memiliki nilai tinggi.

Pada studi ini item dalam daftar periksa dapat dikecualikan apabila item tersebut tidak mendapatkan poin dengan nilai yang relevan pada daftar pemeriksaan, hal ini dibuat agar dapat menemukan literatur dengan ulasan yang sangat relevan dengan studi kasus. Daftar pemeriksaan ini dibuat bukan untuk melakukan perankingan terhadap kualitas dari literatur; akan tetapi tujuannya adalah mendapatkan literatur yang paling sesuai agar dapat digunakan dalam pembahasan studi ini.

S/No	Item Pernyataan	S	CS	TS
1	Apakah penyampaian pernyataan rumusan masalah disampaikan dengan jelas?	Ya	1	Tidak
2	Apakah gab serta kontribusi dari studi telah dinyatakan dengan jelas?	Ya	1	Tidak
3	Apakah hasil studi terkonsentrasi pada sebuah hubungan tunggal serta dapat divalidasi?	Ya	1	Tidak
4	Apakah limitasi serta rekomendasi untuk studi kedepan telah dinyatakan dengan jelas?	Ya	1	Tidak
5	Apakah fokus implementasi DevOps serta prinsip penggunaan telah dinyatakan dengan jelas	Ya	1	Tidak

Tabel 2. Analisa Kesesuaian Literatur

Evaluasi kualitas menggunakan daftar pemeriksaan melalui analisa kesesuaian literatur dalam studi ini, merupakan bentuk ulasan utama yang telah dipilih dari tahapan sebelumnya dengan tujuan untuk mencapai kualitas minimum dari literatur yang akan digunakan. Sedangkan literatur dengan hasil evaluasi tidak sesuai akan didiskualifikasi dari studi ini. Literatur dengan kualitas hasil evaluasi sesuai merupakan literatur yang akan dipertimbangkan untuk

digunakan sebagai referensi utama dalam studi ini. Studi ini akan dilakukan review oleh seorang peneliti yang telah diberikan tugas.

c. Pencarian Literatur

Pencarian literatur merupakan proses yang dilakukan untuk menemukan literatur – literatur sebelumnya yang masih relevan dengan studi pengembangan perangkat lunak. Pada studi ini akan memanfaatkan beberapa sumber elektronik diantaranya, sebagai berikut:

SCOPUS,
ScienceDirect,
Google Scholar,
SpingerLink,
EBSCO,
dan Wiley Online Library

Dalam menemukan literatur yang tepat, maka proses pencarian pada studi ini akan dievaluasi berdasarkan judul, abstrak, dan kata kunci. Literatur dalam Bahasa Inggris dan telah dipublish dari tahun 2019 sampai 2023 yang akan dijadikan bahan pertimbangan sebagai referensi pada studi ini. Hasilnya ditemukan kalimat pencarian umum yang akan digunakan untuk mengeksplorasi secara luas, kalimat yang akan digunakan dalam studi ini adalah “Implementasi DevOps”, “tantangan Devops”, “pengembangan atau DevOps”, dan “pemanfaatan DevOps”. Literatur akan diambil merupakan hasil serupa dari database perpustakaan digital. Studi menggunakan suatu hubungan dalam kaliman pencarian disebabkan karena fokus dari studi ini mengarah pada hasil Analisa data tertentu yang digunakan untuk mendapatkan hasil yang relevan dan bermutu.

d. Seleksi Literatur Berdasarkan Studi Kasus

Tahap awal dalam melakukan seleksi literatur pada studi ini adalah dengan proses membaca judul, abstrak, dan kata kunci kemudian akan diseleksi berdasarkan kriteria pada Tabel 3. Selanjutnya, studi ini akan mendiskualifikas literatur yang tidak sesuai dengan studi kasus dan tidak meninjau literatur yang isisnya kurang dari empat halaman (misalnya, tesis dan abstrak). Pada studi ini literatur seperti buku, judul buku, dan monograf akan dikecualikan, hal ini dilakukan karena biasanya berisi karya penelitian yang sudah dibahas secara lengkap serta hasilnya merupakan penggabungan jurnal atau konfrensi yang telah disajikan atau digunakan oleh literatur sebelumnya.

Dalam menyeleksi literatur hasil analisis kesesuaian literatur juga akan digunakan, kemudian akan dilanjutkan dengan membaca keseluruhan dari literatur yang akan digunakan. Analisa kesesuaian literatur akan didefenisikan pada daftar item yang berisi lima kriteria (terdapat pada tabel 2). Sedangkan literatur yang dinyatakan diterima dan ditolak masing-masing akan diberikan skor oleh evaluasi studi secara individu dengan kriteria “ Ya“ atau “ Tidak“. Analisa kesesuaian literatur dilakukan dengan maksud untuk memisahkan literatur dengan kualitas rendah yang fokus literatur kurang sesuai pada studi ini. Terdapat dua tahapan dalam melakukan Analisa ini. Tahap pertama, literatur yang telah diseleksi akan dibaca oleh peneliti yang telah

ditugaskan berdasarkan kriteria pada Tabel 2. Kriteria “ Ya“ atau “ Tidak“ disediakan dalam proses penerimaan hasil seleksi literatur, literatur dengan kriteria “ Ya“ akan digunakan pada studi ini sedangkan kriteria “ Tidak“ akan didiskualifikasi. Tahap kedua akan dilakukan dengan memastikan bahwa literatur memiliki kualitas tinggi yang telah diterima setelah dilakukan seleksi pada tahap pertama.

e. Pencarian Literatur menggunakan Snowballing

Pencarian menggunakan SnowBalling [32] merupakan Teknik pencarian yang digunakan untuk menentukan literatur yang relevan dengan studi ini. Pemanfaatan SnowBalling digunakan untuk mengidentifikasi konsep tambahan yang mungkin dilewatkan oleh seleksi awal pencarian secara terkomputerisasi. Pada studi ini pendahuluan yang akan diseleksi terlebih dahulu, adopsi SnowBalling dari mundur dari daftar literatur, dan SnowBalling maju yang dikutip dari google scholar (<http://scholar.google.com/>). Pencarian literatur pada studi ini akan dipertimbangkan menggunakan semua literatur yang telah dipublish dari 2019 – 2023 yang tidak termasuk pada tahap awal seleksi literatur. Kemudian, akan diterapkan kedalam seleksi berdasarkan Analisa kesesuaian literatur berdasarkan studi primer lain yang akan digunakan.

f. Klasifikasi Literatur

Studi ini akan mengklasifikasi literatur berdasarkan evaluasi kriteria dan akan diterapkan kedalam model framework yang telah diklasifikasi oleh studi literatur DevOps yang berbeda. Model framework akan diterapkan kedalam tinjauan studi ini secara bertahap. Pertama, draf penjadwalan topik akan didapat berdasarkan judul yang dibaca, kata kunci, abstrak selama melakukan tahap seleksi literatur. Dalam studi ini akan dimasukkan kedalam enam area dengan kemungkinan cabang-cabangnya, kemudian akan dilanjutkan dengan menggunakan label sesuai topik untuk mengklasifikasikan literatur dengan membaca keseluruhan teks serta terus memberikan subtopik terbaru yang akan disesuaikan dengan kebutuhan pada setia area. Terakhir, hasil dari model framework akan dimanfaatkan sebagai bentuk kontribusi berarti untuk tren studi DevOps selama menganalisa data.

Gambar 3 merupakan defenisi dari implementasi DevOps secara berkelanjutan yang diperlukan dalam praktik dengan kualitas terbaik dalam siklus pengiriman. Siklus ini terdiri dari perencanaan, pengembangan, pengujian, penerapan, rilis, dan pemantauan. Sedangkan kualitas kerja sama aktif pada anggota tim yang berbeda terdiri dari serangkaian aktivitas misalkan continuous development, continuous automated testing, deployment, continuous monitoring, integration sustainable, and sustainable delivery serta pedoman tertinggi dengan singkatan “CAMS” (culture, automation, measurement, and sharing).

Gambar 3.1 Model Framework DevOps Loop-Flow

Gambar 3 merupakan model framework aliran Loop mencirikan pedoman yang relevan untuk mengimplementasikan DevOps dan mengilustrasikan pemahaman Continuous Integration dengan fungsi sebagai titik awal dalam pengidentifikasian secara umum dalam mengadopsi DevOps dalam pengetahuan secara luas. Dalam studi ini model framework dibentuk dalam sistem secara perulangan karena disesuaikan dengan proses DevOps yang penerapan proses dilakukan secara berkelanjutan serta berulang di sepanjang pipeline. Untuk mendapatkan pemahaman lebih lanjut mengenai tahapan dari model framework sistem perulangan, maka akan diulas sebagai berikut:

1. Pengembangan Konsep

Konsep dalam studi ini akan dikembangkan berdasarkan enam area model framework aliran perulangan DevOps yang topiknya diambil sesuai dengan konsep dari implementasi DevOps. Literatur yang digunakan menyajikan perspektif dari semua bidang studi yang relevan dengan implementasi DevOps. Pembahasan dari literatur berisi definisi dan metode implementasi DevOps, praktik, pedoman, strategi, manfaat, tantangan, dan lingkup pengembangan perangkat lunak yang masih berhubungan dengan adopsi dan studi selanjutnya dalam mengimplementasikan DevOps dalam tata Kelola TI.

2. Adopsi Metode DevOps

Tahapan pada studi ini akan dilakukan dengan mengkategorikan pendekatan metodologis dalam mengimplementasikan DevOps, seperti model dan otomasi DevOps.

3. Praktik DevOps

Studi ini akan memberikan pemahaman lebih jelas mengenai metode terbaik serta akan memberikan solusi dalam menghadapi beberapa tantangan dalam mengimplementasikan DevOps pada pengembangan perangkat lunak. Kemudian akan dilakukan analisa lebih lanjut mengenai definisi dari metrik DevOps dan strategi dalam mengimplementasikan DevOps seperti configuration management, continuous testing, continuous monitoring, continuous integration, and continuous deployment/delivery.

4. Pedoman DevOps

Tahapan implementasi DevOps akan disebutkan pada studi ini, seperti continuous delivery, continuous deployment, dan continuous integration. Pendekatan pipeline dalam mengimplementasikan DevOps akan digunakan untuk menghasilkan kinerja yang memiliki keandalan, otomatisasi, pengukuran, berbagi umpan balik, dan rilis perangkat lunak yang cepat, seras beberapa hal lainnya.

5. Strategi DevOps

Tahapan ini akan digunakan untuk mengevaluasi berbagai aktivitas dari tim DevOps dan berbagai

pendekatan pada manajemen. Penargetan dari silo dan hasil akan memberikan dukungan serta tanggapan kepada setiap kelompok, analisa proses, dan evaluasi penjaminan kualitas dari pengembangan perangkat lunak dengan sudut pandang berbeda. Tahapan ini juga akan menyajikan penentuan strategi, pemahaman persyaratan, dan pemilihan serta pengujian tools-tools dari DevOps.

6. Manfaat DevOps

Studi ini akan memberikan pemahaman mengenai manfaat mengimplementasikan DevOps dengan menggunakan perangkat yang otomatis bagi tim pengembang dan operasi didalam memecahkan silo yang timbul dari setiap lingkup elemen dari perangkat lunak tertentu.

7. Tantangan DevOps

Tahapan ini akan menyajikan tantangan implementasi DevOps dari berbagai segi berdasarkan kategori literatur. Cangkupan tantangan DevOps akan mengalami perbedaan, hal ini dipengaruhi oleh kebijakan negara, kurangnya standar metodologi, dan kurangnya framework untuk mengadopsi DevOps.

IV. HASIL DAN IMPLEMENTASI

Hasil dari seleksi literatur mendapatkan referensi yang akan digunakan sebagai tinjauan literatur utama. Jumlah literatur yang telah dipilih akan menyajikan beberapa hasil analisa kuantitatif, Tabel 4 akan menunjukkan hasil tersebut.

Indikator (S)	Studi	Referensi	Pemetaan berdasarkan RQ
S1		[1]	RQ1 & RQ3
S2		[2]	RQ3
S3		[3]	RQ1 & RQ2
S4		[4]	RQ1
S5		[5]	RQ1 & RQ2
S6		[6]	RQ12
S7		[7]	RQ1 & RQ3
S8		[8]	RQ1 & RQ2
S9		[9]	RQ1 & RQ2
S10		[10]	RQ1 & RQ2
S11		[11]	RQ1 & RQ2
S12		[12]	RQ1 & RQ2
S13		[13]	RQ3
S14		[14]	RQ3
S15		[15]	RQ1
S16		[16]	RQ1
S17		[17]	RQ1 & RQ2

S18	[18]	RQ1 & RQ2
S19	[19]	RQ2
S20	[20]	RQ2
S21	[21]	RQ2
S22	[22]	RQ1
S23	[23]	RQ1
S24	[24]	RQ1
S25	[25]	RQ1
S26	[26]	RQ3
S27	[27]	RQ3
S28	[28]	RQ3
S29	[29]	RQ1 & RQ2
S30	[30]	RQ1 & RQ2
S31	[31]	RQ2
S32	[32]	RQ2
S33	[33]	RQ2
S34	[34]	RQ1S
S35	[35]	RQ1 & RQ3

Tabel 3

Pemetaan Studi Utama Berdasarkan RQ

Hasil selanjutnya akan diekstrak berdasarkan topik dan pertanyaan penelitian. Sedangkan bagi adopsi implemetasi DevOps dalam mengembangkan perangkat lunak akan diidentifikasi berdasarkan enam topik serta akan diklasifikasi kedalam 3 bagian yang akan dipertimbangkan berdasarkan setiap pertanyaan yang telah dibentuk sebelumnya. Literatur akan dipilih berdasarkan relevansi dan dampaknya, sehingga dapat menjawab pertanyaan penelitian serta dapat dijadikan referensi dengan kualitas yang baik dan sesuai dengan topik yang akan dibahas. Tabel 4 akan menunjukkan literatur yang telah dipilih berdasarkan pertanyaan penelitian.

Topik	Indikator Literatur
Implementasi DevOps	
<i>Pipeline CI/CD</i>	S23, S33, S28, S27, S26, S24, S22, S19, S18, S17, S13, S12, S11, S10, S9, S8, S4, S2
Pemantauan	S23, S32, S28, S27, S26, S19, S17, S10, S9, S4
Otomatisasi	S32, S28, S19, S18, S17, S12, S10, S8, S4

Pendekatan Implementasi DevOps	
Latihan dan Gabungan CI/CD	S5, S15, S20, S6, S14, S16, S7, S21, S25, S29, S30, S31, S34, S35
Strategi	S23, S33, S32, S26, S24, S19, S18, S17
Pedoman, ruang lingkup dan Prinsip	S23, S33, S28, S27, S26, S24, S22, S19, S18, S17, S13, S12, S11, S10, S9, S8, S4, S2, S1, S32, S3, S5, S15, S20, S6, S14, S16, S7
Tantangan Implementasi DevOps	S35, S33, S32, S31, S25, S22, S17, S14, S30, S12, S9
Manfaat Implementasi DevOps	S32, S19, S18, S17, S9

Tabel 4

Identifikasi Literatur berdasarkan Topik

A. Seleksi Literatur

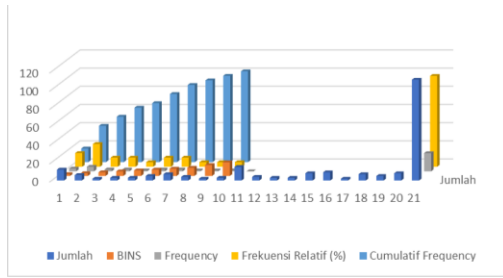
Dalam melakukan seleksi literatur telah dijelaskan pada bagian II metodologi penelitian. Hasilnya 111 literatur didapat dari perpustakaan digital, kemudian akan evaluasi pada tabel 3 dan akan dipilih sebagai literatur yang akan digunakan dengan kriteria yang telah pada tabel 2. Dari hasil evaluasi tersebut maka dipilih 35 refrensi dengan kriteria sesuai dan cukup sesuai dipilih.

Nama Jurnal	Jumlah	BI NS	Freq uenc y	Fre kue nsi Rel atif (%)	Cum ulatif Freq uenc y
Informati on and software teknology	12	2	3	15	15
journal of king saud university and informati on	6	3	5	25	40

sciences					
Nama Jurnal	Jumlah	BI NS	Freq uenc y	Fre kue nsi Rel atif (%)	Cum ulatif Freq uenc y
journal of jilin university	2	4	2	10	50
internatio nal center for research and resources developm ent	3	5	2	10	60
jurnal pendidika n dan teknologi indonesia	3	6	1	5	65
institute of informati on managem ent	5	7	2	10	75
internatio nal journal of computers	7	8	2	10	85
journal of atex class files	4	9	1	5	90
Internatio nal Journal For Research in applied science and engineeri ng technolog y	2	12	1	5	95

Electronic s - MDPI	3	15	1	5	100
IEEE	15				
Software Evolution and Process	4				
A journal for New Zealand Herpetology	3				
IJARSCT	3				
Requirements Engineering	8				
Empirical Software Engineering	9				
Information Technology and Management	2				
Software Quality Journal	7				
Internatio nal Journal of Educational Technology in Higher Education	5				
Sensors - MDPI	8				
	111		20	100	

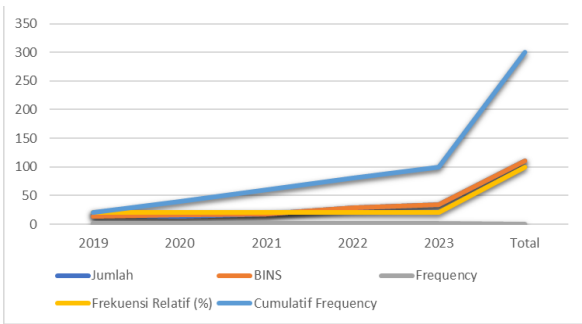
Tabel 5
Pencarian Literatur Berdasarkan Nama Jurnal



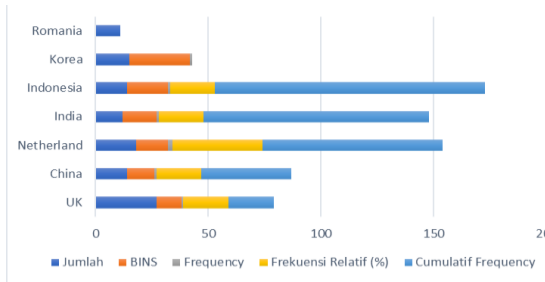
Gambar 4 Pencarian Literatur Berdasarkan Nama Jurnal

Jumlah	BINS	Frequency	Frekuensi Relatif (%)	Cumulatif Frequency
14	1	20	20	20
16	1	20	20	40
18	1	20	20	60
29	1	20	20	80
34	1	20	20	100
111	5	100	100	300

Tabel 6 Pencarian Literatur Berdasarkan Tahun Terbit



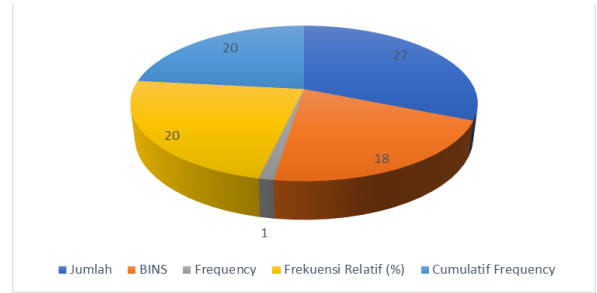
Gambar 5 Pencarian Literatur Berdasarkan Tahun Terbit



Gambar 6 Pencarian Literatur Berdasarkan Negara

No	Jenis Penelitian	Jumlah	BINS	Frequency	Frekuensi Relatif (%)	Cumulatif Frequency
1	Kuantitatif	27	11	14	20	20
2	Kualitatif	18	14	26	20	40
3	Penggabungan kuantitatif dan kualitatif	41	29	26	20	60
4	Teoritis	111	111	4	20	80
	Total	111	111	4	80	200

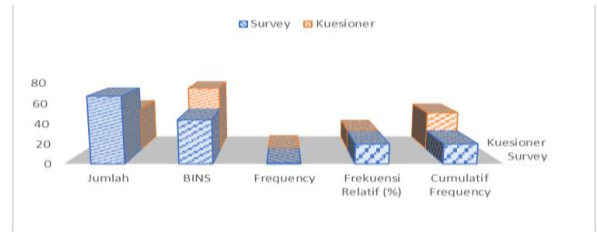
Tabel 8 Pencarian Literatur berdasarkan Jenis Penelitian



Gambar 7 Pencarian Literatur berdasarkan Jenis Penelitian

No	Alat	Jumlah	BINS	Frequency	Frekuensi Relatif (%)
1	Survey	67	44	1	20
2	Kuesioner	44	67	1	20
	Total	111	111	2	40

Tabel 9 Pencarian Literatur berdasarkan Alat Pengumpulan Data



Gambar 8 Pencarian Literatur berdasarkan Alat Pengumpulan Data

BINS	Frequency	Frekuensi Relatif (%)	Cumulatif Frequency
11	1	20	20
14	1	20	40
26	1	20	60
29	1	20	80
111	0	100	300

Tabel 10 Pencarian Literatur berdasarkan banyaknya Peneliti



Gambar 9 Pencarian Literatur berdasarkan banyaknya Peneliti

B. Pedoman Implementasi DevOps Dalam Pengembangan Perangkat Lunak (RQ1)

Dalam mengimplementasi DevOps akan menemukan beberapa hambatan karena dalam proses ini belum memiliki pendekatan atau metode secara langsung. Setiap organisasi ingin mengimplementasikan DevOps dengan prosedur yang berbeda akan tetapi bisa diselaraskan dengan tujuan dari organisasi masing-masing dengan memperhatikan tingkat kematangan DevOps mereka pada saat ini. Pada bagian ini akan diberikan beberapa panduan serta saran umum berkaitan dengan strategi, pendekatan, dan alat dalam mengimplementasikan DevOps dengan memperhatikan studi utama yang telah ditentukan [32].

1. Pipeline Siklus Pengembangan Perangkat Lunak di lingkungan DevOps

Studi [S23, S33, S28, S27, S26, S24, S22, S19, S18, S17, S13, S12, S11, S10, S9, S8, S4, S2] telah membahas bagaimana peningkatan perangkat lunak dengan mengimplementasikan DevOps. Sasaran dari semua studi adalah dapat mencapai kinerja yang lebih baik dalam melakukan pengembangan maupun dalam operasinya. Dalam Studi [S23] menunjukan bahwa dalam mengidentifikasi faktor penentu keberhasilan perangkat lunak maka dibutuhkan pemahaman yang kompleks tentang bagaimana pedoman dalam mengimplementasikan DevOps. Oleh karena itu, pada studi ini dilakukan tinjauan secara sistematis pada literatur sebelumnya tentang bagaimana pedoman dan langkah pengimplementasian DevOps. Hasilnya setelah dilakukan analisa terhadap 38 literatur, maka 100 pedoman faktor keberhasilan dapat diidentifikasi

dan kemudian dapat dikategorikan serta gambaran bagaimana pedoman faktor keberhasilan dapat mempengaruhi dalam mengimplementasikan DevOps. Hal serupa dijelaskan pada studi [S33] dengan mengimplementasikan DevOps dalam mengembangkan perangkat lunak, maka dapat meminimalkan kesenjangan antara pengguna, pengembang, dan operasi. Sedangkan berdasarkan studi [S27] terdapat 3 fase atau pedoman yang harus diterapkan dalam mengimplementasikan pada DevOps: fase pengujian, fase penyebaran, dan fase operasi. Selain itu adapula beberapa penggabungan pedoman yang diterapkan seperti *configuration management, continuous testing, continuous monitoring, continuous integration, and continuous deployment/delivery*.

Studi [S18, S17, S13, S11, S10, S9, S8] mengungkapkan bahwa *continuous integration, and continuous deployment/delivery (CI/CD)* lebih banyak ditekankan pada kecepatan melakukan perilisan perangkat lunak dari ditemukannya perubahan yang tidak terlalu kompleks, incremental serta dalam pemanfaatan otomatisasi selama dilakukannya proses pengembangan. Dalam mengembangkan perangkat lunak maka siklus CI/CD dapat mencakup beberapa elemen, yaitu perencanaan, desain, pengembangan, koreksi kesalahan, pengujian, kinerja, keamanan serta implementasi kedalam lingkup produk. CI adalah sebuah praktek dalam mengembangkan perangkat lunak dimana tim lebih teratur dalam mengintegrasikan pekerjaan. Sebuah organisasi tidak mungkin dapat sekaligus mengadopsi *pipeline* CI/CD. Hal pertama yang dilakukan adalah melatih CI kemudian diadopsi ke CD. Saat terjadi perubahan dari CI ke CD kemudian CD ke CD, maka dapat mengurangi proses eksekusi manual yang kemudian menjadi proses perubahan secara otomatis. Hal serupa juga dijelaskan pada [S17, S13] siklus *pipeline* CI/CD lebih mempercepat rilis dan melakukan evaluasi terhadap perangkat lunak dengan kualitas yang lebih baik serta dapat memenuhi target. Sedangkan studi [S10] menunjukan tiga pedoman utama yang harus diperhatikan dalam mengimplementasikan DevOps melalui *pipeline* CI/CD yaitu: pertama implementasi *pipeline* merupakan proses yang dilakukan untuk menerapkan alur perancangan dengan memperhatikan Langkah-langkah seperti (1) persiapan struktur, (2)

menentukan alur kerja pengembangan skrip, (3) pemilihan alat migrasi. Kedua, analisis kuantitatif yaitu proses pengukuran terhadap keuntungan yang akan dihasilkan oleh tim pengembang. Ketiga, analisis kualitatif merupakan prosedur evaluasi sebelum dan setelah mengimplementasikan *pipeline* CI/CD. Studi [S8] proses *pipeline* CI/CD sangat bertanggung jawab terhadap proses otomatis dalam melakukan pengiriman perangkat lunak. Adapun pedoman yang harus diperhatikan seperti: budaya kolaboratif tim, praktik secara Kontinu, analitik DevOps, arsitektur layanan mikro, dan rilis perangkat lunak. Sedangkan pada studi [S9] mengungkapkan bahwa DevOps dapat diimplementasikan ke dalam perangkat lunak IoT melalui kumpulan siklus, alat, dan layanan CI/CD.

Studi [S26, S24, S22, S19] menunjukkan bagaimana menerapkan pedoman dalam mengimplementasikan DevOps dapat menjembatani antara pengembangan dan operasi pada kualitas perangkat lunak. Dalam studi ini ditunjukkan empat pola dasar DevOps: *version control everything, continuous integration, deployment automation, dan monitoring*. Selain itu, dalam mengimplementasikan DevOps secara kontinu dalam pengembangan dan operasi maka dibutuhkan perubahan dalam aspek *culture, automation, measurement, dan share* (CAMS). Aspek *Culture* menjadi prioritas dalam mengimplementasikan DevOps, karena sangat berpengaruh terhadap kualitas dalam menyediakan layanan. *Automation* merupakan aspek yang menghasilkan suatu standar serta hasil yang dapat diprediksi untuk meningkatkan proses *delivery* dari perangkat lunak. Aspek *measurement* digunakan sebagai proses dalam memberikan penilaian dan pengukuran yang sistematis sehingga terjadi peningkatan secara kontinu. Aspek *share* digunakan untuk berbagi pengetahuan diantara tim yang terdapat dalam organisasi, sehingga dapat mendukung penerapan DevOps yang berhasil. Studi [S13, S4, S2] mengidentifikasi bagaimana siklus hidup pengembangan DevOps dengan menerapkan praktik *pipeline* CI/CD. Tujuan utama dari mengembangkan perangkat lunak modern adalah membangun, menguji dan rilis dengan cepat dan

andal. Oleh karena itu, pedoman *pipeline* CI/CD seperti sumber, pembuatan, pementasan, dan produksi harus diterapkan oleh tim pengembang.

2. Controlling

Studi [S23, S32, S28] menjelaskan bahwa controlling atau pemantauan merupakan tahapan lainnya yang penting dalam siklus kelangsungan DevOps. Dalam tahapan ini pengembang harus terus melakukan pemantauan terhadap kinerja perangkat lunak. Pada tahap ini akan dilakukan monitoring terhadap semua informasi awal yang berkaitan dengan sistem perangkat lunak, hal ini akan mempercepat pemrosesan informasi utama dalam mengenali perangkat lunak dengan cepat. Pemantauan dibagi kedalam dua kategori utama: pemantauan infrastruktur, pemantauan keamanan dan pemantauan otomatisasi yang bertujuan untuk mengotrol status dan fungsionalitas sistem serta dapat menunjukkan kepada pengembang apabila terjadi ketidaknormalan dalam sistem. Sedangkan studi [S27] berpendapat bahwa penerapan pemantauan dengan memanfaatkan siklus PDCA dapat memfasilitasi integrasi antara alat serta proses dalam melakukan pengembangan. Oleh karena itu, penerapan pemantauan perangkat lunak secara komperhensif dapat memungkinkan tim dalam menentukan Kesehatan serta status perangkat lunak. Selain itu, tersedianya alat yang memungkinkan tim dapat secara aktif melakukan debug dan pengamatan terhadap perangkat lunak [S26].

Studi [S10, S9] menyebutkan bahwa dalam mengoptimalkan kemampuan untuk melakukan pemantauan dan pengelolaan terhadap kinerja serta ketersediaan infrastruktur perangkat lunak secara terus menerus, maka praktik pemantauan harus memperhatikan beberapa hal: pemantauan otomatis, mengukur metrik kinerja utama, memantau kinerja anggota dari tim, menganalisa resiko serta kebutuhan keamanan. Sedangkan studi [S4] memperhatikan kontrak cerdas dalam melakukan pemantauan merupakan tugas yang menantang dalam memberikan informasi yang penting dari pengembangan perangkat lunak.

3. Tools

Studi [S32, S28, S19, S18, S17, S12, S10, S8, S4] merupakan studi utama yang menunjukkan alat sebagai faktor utama dalam mengimplementasikan DevOps. Alat DevOps adalah elemen penting dalam proses melakukan pengembangan dalam *pipeline* CI/CD. DevOps dapat diterapkan melalui penggabungan proses, orang, dan alat-alat. Otomatisasi DevOps merupakan alat tambahan dalam mendapatkan *feedback* dari pengembangan dan operasi secara continue. Alat SCM membantu melakukan penyimpanan sintaks dalam repositori, pembuatan versi sintaks, dan membantuk mendistribusikan perangkat lunak. Jenkins merupakan alat serbaguna dalam menjalankan tes oleh tim pengembang secara otomatis. *Flyway* merupakan alat dalam melakukan migrasi database yang paling populer. *Utsql* merupakan alat pengujian otomatis database dalam unit kerangka pengujian. Alat lain seperti *dbunit*, *jUnit*, *parser SQL Postgres*, *SQLFluff* dan *codecop* digunakan untuk pengujian, perpanjangan, pemeriksaan sintaks, kerentanan, serta pemeriksaan dalam perangkat lunak. *Slack dan Hip Chat* digunakan sebagai alat komunikasi DevOps dengan memanfaatkan pendekatan melalui pesan teks dalam melaporkan melalui notifikasi peringatan dalam melakukan pemantauan. *GitHub* dan *Bitbucket* merupakan alat yang dipakai dalam melakukan kontrol versi yang dapat membantu mendorong perubahan ke pusat repositori. Sedangkan alat pemantauan otomatis serta dapat memberikan pesan error dalam bentuk grafik dapat menggunakan *new reluk*, *graylog*, dan *kinesis*. *Ansible*, *Chef*, *Jenkins*, *puppet*, dan *docker* dapat digunakan untuk otomatisasi, containerisasi, pengiriman dan penerapan setelah selesai dilakukan pengujian. Alat-alat ini juga sangat memungkinkan melakukan *continuous integration* dalam *pipeline* CI/CD pada saat mengimplementasikan DevOps pada pengembangan perangkat lunak.

4. Strategi Implementasi

Studi [S23, S33, S32, S26, S24, S19, S18, S17] menjelaskan bahwa implementasi DevOps telah menjadi salah satu strategi yang banyak digunakan dalam mengembangkan perangkat lunak. Dalam mengembangkan perangkat lunak, DevOps memiliki *pipeline* CI/CD yang dapat

memudahkan melakukan penggabungan antara tim pengembang dan operasi pada organisasi serta dapat melakukan pengiriman program yang efektif. Pemodelan bisnis, analisis, pemrograman, instalasi, desain, pengiriman program, pemantauan, operasi, dan evaluasi merupakan beberapa strategi yang dapat digunakan dalam fase pengembangan perangkat lunak. Sedangkan pemantauan manajemen, pengontrolan versi, alat otomatisasi, dan penerapan merupakan tahapan integrasi dalam melakukan penyederhanaan proses pada penetapan pedoman dan dapat menentukan strategi pengembang yang sesuai.

C. Tantangan Implementasi DevOps Dalam Pengembangan Perangkat Lunak (RQ2)

Dalam mengimplementasikan DevOps pada pengembangan perangkat lunak berdasarkan hasil tinjauan dari beberapa literatur studi [S35, S33, S32, S31, S25, S17, S14, S30, S12, S9] ditemukan bahwa tidak semua pemanfaatan DevOps dapat mencapai keberhasilan. Hal ini bisa dipengaruhi karena ketidakmampuan organisasi dalam menghadapi tantangan dari DevOps. Oleh karena itu, pada studi ini akan diberikan beberapa tantangan yang bisa ditemui dalam pemanfaatan DevOps. Adapun beberapa tantangannya adalah sebagai berikut:

1. Kurangnya tenaga terampil dengan kemampuan teknis yang diperlukan.
2. Adanya penolakan dan ketidakpastiaan dalam melakukan perubahan dari budaya sebelumnya.
3. Kesulitan mengubah tumpukan alat dan teknologi.
4. Belum ditetapkannya standar pedoman.
5. Hambatan terhadap perubahan dan resistensi.
6. Kekurangan biaya serta infrastruktur.
7. Ketidakjelasan dalam mendefinisikan konsep.
8. Perlu memvalidasi metrik tolak ukur.
9. Kurangnya otomatisasi.
- 10.Keamanan.
- 11.Pengujian.
- 12.Integrasi.
- 13.Infrastruktur.

Berdasarkan beberapa tantangan yang telah diberikan, maka organisasi wajib menentukan strategi dalam mengatasi tantangan secara tepat sebelum mengimplementasikan DevOps. Beberapa langkah dasar yang dapat digunakan perusahaan [S25, S17] seperti : (1) memberikan pelatihan dan pendidikan yang tepat, (2) melibatkan karyawan dalam proses

pengambilan keputusan, (2) menjelaskan manfaat implementasi DevOps secara jelas, (3) mengembangkan budaya kolaborasi dan tanggung jawab bersama, (4) merencanakan transmisi secara berhati-hati. Langkah strategis lain bisa diciptakan organisasi dengan memperhatikan topik perangkat lunak yang akan dikembangkan.

D. Manfaat Implementasi DevOps Dalam Pengembangan Perangkat Lunak (RQ3)

Dalam mengembangkan perangkat lunak dengan mengimplementasikan DevOps berdasarkan literatur sebelumnya [S32, S19, S18, S17, S9] mengemukakan bahwa organisasi akan memperoleh berbagai manfaat. Namun, organisasi wajib untuk memahami secara luas tentang manfaat DevOps. Dalam studi ini, setelah dilakukan analisa lebih lanjut pada lima literatur, maka manfaat dapat dikategorikan sebagai berikut:

1. Siklus release mengalami peningkatan kecepatan pada waktu penyelesaian kode serta mengirim kode ke produksi.
2. *Throunghput* tinggi (output)
3. Meningkatkan kualitas tim dan kualitas kode.
4. Meningkatkan kepuasan pelanggan.
5. Meningkatkan kolaborasi dan komunikasi antar tim.
6. Proses pengembangan perangkat lunak lebih efisien.
7. Memberikan keunggulan kompetitif kepada organisasi serta dapat membantu organisasi tetap fleksibel, mudah beradaptasi, dan efektif terhadap perubahan.
8. *Pipeline CI/CD* mengubah hasil pengujian dengan pengetahuan.
9. *Pipeline CI/CD* menjamin kualitas perangkat lunak.
10. *Pipeline CI/CD* dapat mengukur kesehatan perangkat lunak.
11. *Pipeline CI/CD* mengurangi *overhead* pada biaya.
12. *Pipeline CI/CD* meningkatkan waktu perilsan dan pengujian.
13. *Pipeline CI/CD* memberikan kumpulan pengembangan perangkat lunak yang lebih bervariasi.
14. *Pipeline CI/CD* memungkinkan pengembang dapat mengidentifikasi bug dan anomali.

Beberapa manfaat ini memiliki satu tujuan utama yaitu meningkat kualitas perangkat lunak pada organisasi.

E. Ancaman Terhadap Validitas

Validitas digunakan untuk mendapatkan hasil penelitian yang tidak bias atau mengandung kesalahan. Dalam studi ini pertimbangan validitas dimanfaatkan pada saat melakukan seleksi terhadap literatur studi sebelumnya. Tindakan ini meliputi membatasi istilah seleksi literatur, melakukan pemilihan terhadap mesin pencarian yang tepat, memperhatikan hasil ekstraksi data yang tidak akurat, serta menentukan kriteria inklusi dan eksklusi yang dapat menyebabkan bias disaat melakukan seleksi literatur. Dalam memperhatikan ancaman terhadap validitas, maka studi ini akan membahas secara mendetail melalui validitas internal dan eksternal.

Validitas internal pada studi ini digunakan untuk menyatakan bahwa studi mengenai tinjauan sistematis bisa lakukan kembali. Dalam studi ini penentuan *keyword* seleksi serta indikator kriteria inklusi dan eksklusi diperhatikan secara cermat. Dengan memperhatikan *keyword* dan mesin pencarian serta kejadian bias lainnya, maka akan sangat berguna untuk mencegah terjadinya masalah ketidaklengkapan data pada awal melakukan pengumpulan literatur. Studi ini juga akan melakukan identifikasi terhadap literatur yang relevan, menggunakan beberapa *keyword* berbeda pada saat menyeleksi literatur yang masih memiliki keterkaitan dengan studi yang dilakukan, terakhir melakukan seleksi secara manual pada refrensi. Pada studi ini *mendeley* digunakan sebagai database untuk mengontrol ancaman terjadinya ketidakvalid data. Dengan menerapkan strategi ini, maka studi ini dapat dinyatakan bahwa seleksi pada literatur telah memenuhi standar dan inklusif serta menurunkan persentase kehilangan publikasi dari studi. Hal lain yang diperhatikan adalah pengaruh penilaian peneliti terhadap proses seleksi literatur menggunakan kriteria inklusif dan eksklusif. Dengan menerapkan dua kriteria ini, maka studi ini dapat mengurangi bias dalam proses seleksi literatur.

Validitas eksternal pada studi ini dilakukan dengan membatasi proses seleksi pada penulisan dari literatur. Studi ini akan menggunakan dua kategori penulisan literatur yaitu Indonesia dan Inggris.

Sedangkan mesin pencari, database, dan *keyword* pada proses seleksi literatur juga akan dibatasi sehingga bisa mencegah terjadinya bias atau kesalahan. Namun, pembatasan ini menimbulkan pertanyaan apakah studi ini sudah membahas secara mendetail mengenai implementasi DevOps. Oleh karena itu, pada studi ini untuk mendapatkan informasi yang luas dari literatur yang diseleksi, maka akan memanfaatkan *keyword* yang lebih luas serta memanfaatkan database yang paling umum.

V. SIMPULAN DAN SARAN

Studi ini dilakukan untuk meninjau secara sistematis mengenai pedoman, tantangan, serta manfaat dalam mengimplementasikan DevOps untuk pengembangan dan operasi perangkat lunak. Kontribusi utama dari studi ini adalah dapat memberikan pemahaman secara mendetail mengenai implementasi DevOps. Studi dimulai dengan melakukan proses seleksi literatur menggunakan beberapa kriteria seperti nama jurnal, tahun terbit, negara, jenis penelitian, alat ekstraksi data serta banyak author. Literatur yang telah diseleksi kemudian akan direview menggunakan kriteria inklusif dan analisa ketidaksesuaian pada literatur. Hasilnya kami berhasil melakukan seleksi terhadap 35 literatur dengan kriteria sesuai dan cukup sesuai dengan masalah yang akan diidentifikasi secara luas. Berdasarkan 35 literatur tersebut, maka dapat diidentifikasi tiga poin utama yaitu pedoman, manfaat, dan tantangan dalam mengimplementasikan DevOps. Berdasarkan tinjauan yang telah dilakukan, maka ditemukan bahwa pipeline CI/CD dapat diterapkan untuk meningkatkan kualitas, membangun, menguji dan rilis dengan cepat dan andal serta mengembangkan perangkat lunak modern. Oleh karena itu, pedoman pipeline CI/CD seperti sumber, pembuatan, pementasan, dan produksi harus diterapkan oleh tim pengembang. Hal lain seperti pematauan, tools, dan strategi juga harus diperhatikan karena masih termasuk kedalam siklus pedoman implementasi DevOps. Studi ini juga menemukan beberapa manfaat dari DevOps dalam mengembangkan perangkat lunak. Namun, implementasi dapat bermanfaat apabila organisasi dapat mengatasi tantangan pemanfaatan DevOps dalam mengembangkan perangkat lunak. Oleh karena itu, pada studi ini telah diberikan pemahaman tentang bagaimana mengatasi beberapa tantangan dari pemanfaatan DevOps. Untuk penelitian kedepan peneliti berencana akan melakukan tinjauan secara sistematis mengenai

perkembangan DevOps ke DevSecOps, MLOps, dan AIOps dengan memperhatikan pedoman, tantangan, serta manfaat dalam mengembangkan perangkat lunak.

REFERENSI

- [1] Ahmed, S., Singh, M., Doherty, B., Ramlan, E. I., Harkin, K., Bucholc, M., Ahmed, S., Singh, M., Doherty, B., Ramlan, E., Harkin, K., Bucholc, M., & Coyle, D. (n.d.). Knowledge-based Intelligent System for IT Incident DevOps.
- [2] Fritsch, J., Bogner, J., Haug, M., Franco da Silva, A. C., Rubner, C., Saft, M., Sauer, H., & Wagner, S. (2023). Adopting microservices and DevOps in the cyber-physical systems domain: A rapid review and case study. *Software - Practice and Experience*, 53(3), 790–810. <https://doi.org/10.1002/spe.3169>.
- [3] Tanilkan, S. S., & Hannay, J. E. (n.d.). Projects VS Continuous Product Development-Does it Affect Benefits Realization?.
- [4] Reyes, A., Jimeno, M., & Villanueva-Polanco, R. (2023). Continuous and Secure Integration Framework for Smart Contracts. *Sensors*, 23(1). <https://doi.org/10.3390/s2301054>.
- [5] Fernandez-Gauna, B., Rojo, N., & Graña, M. (2023). Automatic feedback and assessment of team-coding assignments in a DevOps context. *International Journal of Educational Technology in Higher Education*, 20(1). <https://doi.org/10.1186/s41239-023-00386-6>.
- [6] Lombardi, F., & Fanton, A. (2023). From DevOps to DevSecOps is not enough. *CyberDevOps: an extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline*. *Software Quality Journal*. <https://doi.org/10.1007/s11219-023-09619-3>.
- [7] Dehghani, R., & Ramsin, R. (2023). A knowledge management-driven and DevOps-based method for situational method engineering. *Information Technology and Management*. <https://doi.org/10.1007/s10799-023-00386-y>.
- [8] Hernández, R., Moros, B., & Nicolás, J. (2023). Requirements management in DevOps environments: a multivocal mapping study. *Requirements Engineering*. <https://doi.org/10.1007/s00766-023-00396-w>.
- [9] Kavita Dhakad. (2023). Adopting Continuous Integration Practices to Achieve Quality in DevOps. *International Journal of Advanced Research in Science, Communication and Technology*, 101–119. <https://doi.org/10.48175/ijarsct-8368>.
- [10] Dhakad, K., Prof, A., & Gholap, P. (n.d.). IMPROVING QUALITY METRICS IN DEVOPS

THROUGH CONTINUOUS INTEGRATION PRACTICES. 2, 2023.

[11] Bijwe, A., & Shankar, P. (n.d.). DEVOPS CULTURE AND PRACTICES FOR IOT APPLICATIONS. 12, 2023.

[12] Fluri, J., Fornari, F., & Pustulka, E. (n.d.). Measuring the Benefits of CI/CD Practices for Database Application Development Teaching Optimization with Games View project Teaching SQL View project Measuring the Benefits of CI/CD Practices for Database Application Development. www.oracle.com.

[13] Bertolino, A., de Angelis, G., Guerriero, A., Miranda, B., Pietrantuono, R., & Russo, S. (n.d.). DevOpRET: Continuous Reliability Testing in DevOps. <http://mc.manuscriptcentral.com/jsme>.

[14] Martina, M. R., Bianchini, E., Sincer, S., Francesconi, M., & Gemignani, V. (2023). Software medical device maintenance: DevOps based approach for problem and modification management. *Journal of Software: Evolution and Process*. <https://doi.org/10.1002/smr.2570>.

[15] Learning DevOps with an Industrial Perspective in the Software Learning DevOps with an Industrial Perspective in the Software Engineering Curricula Engineering Curricula. (n.d.). <https://doi.org/10.36227/techrxiv.22096535.v1>.

[16] Buttar, A. M., Khalid, A., Alenezi, M., Akbar, M. A., Rafi, S., Gumaei, A. H., & Riaz, M. T. (2023). Optimization of DevOps Transformation for Cloud-Based Applications. *Electronics (Switzerland)*, 12(2). <https://doi.org/10.3390/electronics12020357>.

[17] Srivastava, S. (n.d.). DevOps: A New Approach for Bridging the Gap between Development and Operations Teams. <https://www.researchgate.net/publication/370325720>.

[18] Bashiru, O., & Olufemi, O. G. (2023). An Enhanced CICD Pipeline: A DevSecOps Approach. *International Journal of Computer Applications*, 184(48), 8–13. <https://doi.org/10.5120/ijca2023922594>.

[19] Singh, A. (2021). A Comparison on Continuous Integration and Continuous Deployment (CI/CD) on Cloud Based on Various Deployment and Testing Strategies. *International Journal for Research in Applied Science and Engineering Technology*, 9(VI), 4968–4977. <https://doi.org/10.22214/ijraset.2021.36038>.

[20] Cankar, M., Petrovic, N., Pita Costa, J., Cernivec, A., Antic, J., Martincic, T., & Stepec, D. (2023). Security in DevSecOps: Applying Tools and Machine Learning to Verification and Monitoring Steps. *ICPE 2023 - Companion of the 2023 ACM/SPEC Interna-*

tional Conference on Performance Engineering, 201–205. <https://doi.org/10.1145/3578245.3584943>.

[21] Dakkak, A., Bosch, J., & Olsson, H. H. (2023). DevServOps: DevOps For Product-Oriented Product-Service Systems. <http://arxiv.org/abs/2305.08601>.

[22] Ruiz, J. M. S., Mayo, F. J. D., Oriol, X., Crespo, J. F., Benavides, D., & Teniente, E. (2023). A Benchmarking Proposal for DevOps Practices on Open Source Software Projects. <http://arxiv.org/abs/2304.14790>.

[23] Azad, N., & Hyrynsalmi, S. (2023). DevOps critical success factors - A systematic literature review. *Information and Software Technology*, 157. <https://doi.org/10.1016/j.infsof.2023.107150>.

[24] Zohaib, M. (n.d.). Towards Sustainable DevOps: A Decision Making Framework.

[25] Ferino, S., Fernandes, M., Cirilo, E., Agnez, L., Batista, B., Kulesza, U., Aranha, E., & Treude, C. (2023). Overcoming Challenges in DevOps Education through Teaching Methods. <http://arxiv.org/abs/2302.05564>.

[26] Marques, P., & Correia, F. F. (2023). Foundational DevOps Patterns. <http://arxiv.org/abs/2302.01053>.

[27] Dovleac, R. (n.d.). The rise of DevQualOps and implications on software quality. <http://www.iaras.org/iaras/journals/ijc>.

[28] Pando, B., & Dávila, A. (2023). <https://ispranproceedings.elpub.ru/jour/article/view/1612>. *Proceedings of the Institute for System Programming of the RAS*, 35(1), 163–188. [https://doi.org/10.15514/ISPRAS-2023-35\(1\)-11](https://doi.org/10.15514/ISPRAS-2023-35(1)-11).

[29] Niño-Martínez, V. M., Ocharán-Hernández, J. O., Limón, X., & Pérez-Arriaga, J. C. (2023). Microservice Deployment. *Proceedings of the Institute for System Programming of the RAS*, 35(1), 57–72. [https://doi.org/10.15514/ISPRAS-2023-35\(1\)-4](https://doi.org/10.15514/ISPRAS-2023-35(1)-4).

[30] Sánchez-Cifo, M. Á., Bermejo, P., & Navarro, E. (2023). DevOps: Is there a gap between education and industry? *Journal of Software: Evolution and Process*. <https://doi.org/10.1002/smr.2534>.

[31] Afzal, M., waseem Iqbal, M., Haseeb, U., Hameed, U., Zubair Ahmed, S., Waseem Iqbal, M., & Arif, S. (2023). ADOPTION OF CONTINUOUS DELIVERY IN DEVOPS: FUTURE CHALLENGES. <https://doi.org/10.17605/OSF.IO/6NYPX>.

[32] Hamza, U., Syed-Mohamad Sharifah, M., & Lee Abdullah, N. (n.d.). DevOps Adoption Guidelines, Challenges, and Benefits: A Systematic Literature Review. *ICRRD Journal*, 2023(1), 2773–5958. <https://doi.org/10.53272/icrrd>.

[33] Moez, M., Bashir, A., Sardar, M., Professor, A., Zubair Ahmad, S., & Aqeel, M. (2023). ADOPTING A DEVOPS APPROACH TO SOFTWARE DEVELOPMENT AND OPERATIONS: A SURVEY MUHAMMAD WASEEM IQBAL. 42. <https://doi.org/10.17605/OSF.IO/2YPFA>.

[34] Hany Fawzy, A., Wassif, K., & Moussa, H. (2023). Framework for automatic detection of anomalies in DevOps. *Journal of King Saud University - Computer and Information Sciences*, 35(3), 8–19. <https://doi.org/10.1016/j.jksuci.2023.02.010>.

Hrusto, A., Engström, E., & Runeson, P. (2023).