BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Berkembangnya teknologi informasi saat ini, penggunaan arsitektur *Cloud Native* semakin umum dalam pengembangan aplikasi. Pendekatan ini memungkinkan pembangunan sistem yang lebih fleksibel, dapat diperbarui secara mandiri, dan dikembangkan secara cepat[1]. Selain itu, ketersediaan platform cloud untuk deployment aplikasi turut memudahkan para developer. Namun demikian, proses deployment aplikasi *cloud native* tetap menyimpan sejumlah tantangan. Para pengembang aplikasi menggunakan prinsip Devops dalam development aplikasi berbasis cloud native. Devops menggabungkan pengembangan (Dev) dan operasi (Ops) untuk menyatukan orang, proses, dan teknologi dalam perencanaan, pengembangan, pengiriman, dan operasi aplikasi[2].

Continuous Integration/Continuous Deployment (CI/CD) proses CI melibatkan penggabungan kode dan fitur baru ke dalam repositori utama, sementara CD bertanggung jawab untuk melakukan deployment otomatis setelah proses build dan pengujian berhasil[3]. Dalam penerapan proses CD dilakukan dengan metode push based deployment dengan cara melakukan push konfiugarsi ke lingkungan tujuan, namun terdapat beberapa tantangan signifikan yang dihadapi. Salah satu masalah utama adalah lamanya proses rollback aplikasi saat terjadi masalah setelah deployment, yang dapat menyebabkan gangguan berkepanjangan bagi pengguna aplikasi. Ketika terjadi bug atau error pada deployment versi terbaru dari aplikasi epsi maka proses untuk mengembalikannya ke versi sebelumnya cukup lama dengan menggunakan metode ini proses akan membutuhkan waktu sekitar 6-7 menit. Selain itu, adanya akses langsung tim pengembang ke kluster Kubernetes meningkatkan risiko miskonfigurasi dan penyimpangan konfigurasi (configuration drift), yang dapat menyebabkan kegagalan aplikasi dan berdampak negatif pada pengalaman pengguna[4]. Ketika developer ingin melakukan suatu perubahan pada komponen di lingkungan kubernetes, developer akan menerapkan perubahan itu secara langsung pada kluster kubernetes melalui kubectl client. Jika terdapat

kesalahan pada perubahan konfigurasi tersebut maka itu akan berdampak secara langsung pada aplikasi dan pengguna[5].

Pendekatan push-based deployment memiliki beberapa kerentanan keamanan yang perlu diperhatikan. Salah satu masalah utama adalah manajemen kredensial Kubernetes, di mana kredensial (kubeconfig) disimpan dalam proses CI/CD eksternal seperti GitHub Actions[6]. Hal ini berpotensi menimbulkan risiko jika kredensial tersebut terekspos atau disalahgunakan. Penting untuk memastikan kredensial disimpan dengan aman, seperti di tempat penyimpanan rahasia (secret store) yang dienkripsi. Selain itu, push-based deployment juga memiliki kelemahan dalam hal kontrol dan audit trail. Tanpa mekanisme kontrol yang ketat, sulit untuk melacak perubahan dan mengidentifikasi masalah jika terjadi deployment yang tidak diinginkan atau berbahaya[7]. Risiko kesalahan konfigurasi juga lebih tinggi tanpa proses review yang memadai, yang dapat menyebabkan masalah stabilitas atau keamanan pada sistem yang berjalan. Terakhir, metode ini memungkinkan developer untuk mengakses dan melakukan konfigurasi langsung pada kluster Kubernetes, yang dapat meningkatkan risiko keamanan jika tidak dikelola dengan baik. Oleh karena itu, pendekatan push-based deployment memiliki kerentanan keamanan yang signifikan dan dapat menimbulkan risiko operasional yang besar bagi organisasi jika tidak ditangani dengan benar.[8]

Penelitian ini mengambil studi kasus pada aplikasi Electronic Posyandu System Indonesia (EPSI) yaitu aplikasi berbasis mobile dengan fitur forum konsultasi antara bidan desa /dokter dengan orang tua balita untuk menghilangkan kesulitan orang tua untuk mendapatkan bimbingan seputar perawatan dan vitamin untuk bayi, lalu fitur Buku KIA digital melalui aplikasi mobile untuk menghilangkan kesulitan orang tua kesulitan untuk mendapatkan informasi mengenai status pertumbuhan bayi dan fitur Penjadwalan / pengingat untuk orang tua bahwa ada kegiatan posyandu untuk menghilangkan kesulitan pengurus posyandu untuk memberikan informasi kegiatan posyandu kepada orangtua. Aplikasi EPSI ini terdiri dari CMS yang berfungsi sebagai backoffice atau backend aplikasi dimana admin dan petugas posyandu dapat mengelola data, REST API yang berfungsi sebagai komunikator

atau interface untuk pengelolaan data, dan Aplikasi Mobile yang difungsikan untuk para orang tua dapat menggunakan aplikasi ini.

Gitops adalah model operasi untuk aplikasi *cloud native* yang menyimpan aplikasi dan kode infrastruktur deklaratif di Git untuk digunakan sebagai sumber kebenaran untuk pengiriman berkelanjutan otomatis[9]. Dengan *Gitops*, kita dapat menjelaskan status yang diinginkan dari seluruh sistem kita dalam repositori git, dan operator *Gitops* menyebarkannya ke lingkungan gitops dengan mengunkan konsep *pull based deployment* yaitu operator akan melakukan sinkronisasi antara konfigurasi digit ke kluster *kubernetes*. *Argo cd* merupakan salah satu alat untuk menggunakan metode *gitops*, argocd membantu kita untuk mempercepat proses pengembangan aplikasi lalu mempermudah manajemen infrastruktur dan juga meningkatkan efisiensi serta kemanan dalam proses pengembangan aplikasi *cloud native*[10].

Dengan menggunakan metode *Gitops* ini maka masalah deployment aplikasi *cloud native* pada metode *push based deployment* akan terpecahkan. Proses *rollback* aplikasi akan menjadi lebih cepat, mengurangi adanya kesalahan yang dilakukan oleh pengembang dan juga meningkatkan keamanan jalannya proses *deployment*.

1.2 Rumusan Masalah

Permasalahan penelitian yang penulis ajukan ini dapat dirumuskan menjadi rumusan masalah sebagai berikut:

1. Bagaimana menerapkan proses *rollback* aplikasi yang cepat dan efisien agar tidak menimbulkan ketidaknyamanan bagi pengguna?
2. Bagaimana menerapkan konfigurasi yang tepat dan mengoptimalkan proses pengembangan untuk mengurangi terjadinya error pada aplikasi?
3. Bagaimana menerapkan langkah-langkah keamanan yang memadai untuk mencegah celah keamanan dan penyerangan saat proses *deployment* aplikasi?

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah untuk menganalisis sebuah prototype sistem deployment aplikasi menggunakan konsep *Gitops* dan pull based *deployment* dengan membandingkan dengan metode sebelumnya yaitu push based *deployment*. Sedangkan tujuan dari pengembangan metode ini adalah:

1. Menerapkan proses *rollback* aplikasi yang cepat dan efisien agar tidak menimbulkan ketidaknyamanan bagi pengguna.
2. Menerapkan konfigurasi yang tepat dan mengoptimalkan proses pengembangan untuk mengurangi terjadinya error pada aplikasi.
3. Menerapkan langkah-langkah keamanan yang memadai untuk mencegah celah keamanan dan penyerangan saat proses *deployment* aplikasi.

1.4 Batasan Masalah

Adapun Batasan-batasan masalah yang ada di dalam penelitian ini meliputi:

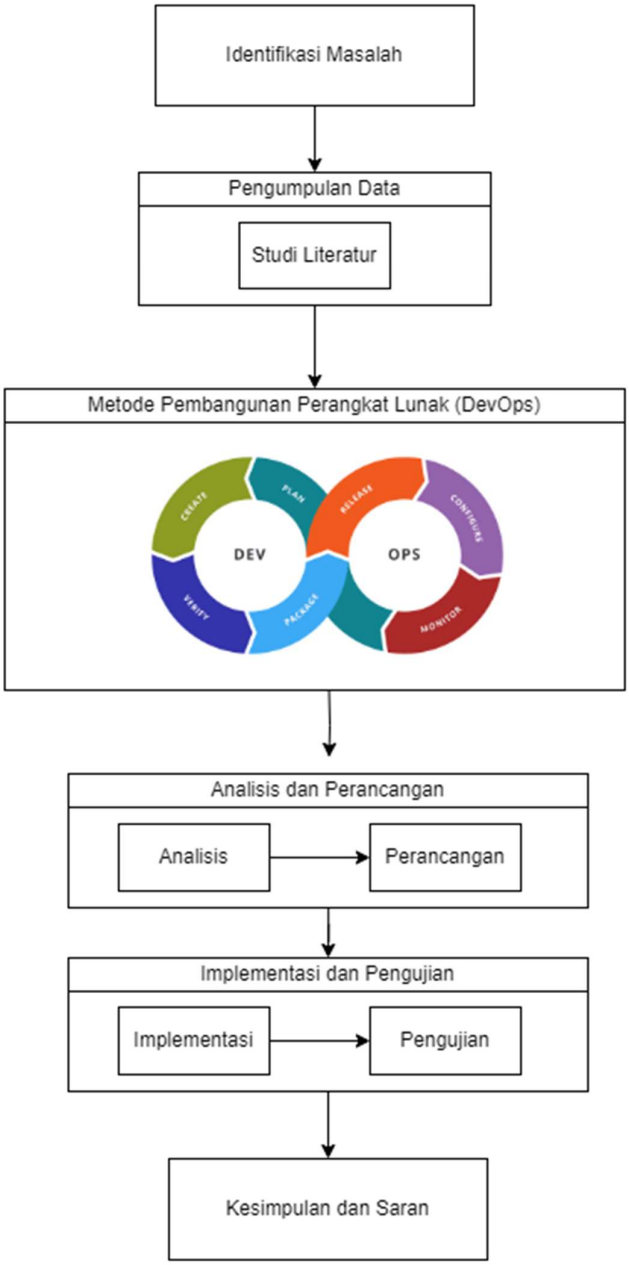
1. Menggunakan argocd sebagai alat *gitops*
2. Lingkungan tujuan menggunakan kubernetes sebagai *container orchestration*
3. Source code management menggunakan github
4. Konfigurasi pipeline menggunakan github action
5. Menggunakan docker sebagai *container system*
6. Hasil dari penelitian berupa rancangan sistem dan *source code* konfigurasi
7. Rancangan sistem hanya berfokus pada konsep *gitops*
8. Menggunakan helm chart sebagai deklaratif konfigurasi service pada *kubernetes*
9. Rancangan sistem hanya dapat diimplementasikan pada aplikasi berbasis cloud native.

10. Menggunakan Vault dan External Secret operator sebagai alat untuk manajemen secret
11. Pengujian dilakukan pada aplikasi Electronic Posyandu System Indonesia (EPSI) API dan EPSI CMS.

1.5 Metodologi Penelitian

Metodologi penelitian merupakan suatu proses untuk memecahkan sebuah permasalahan secara logis, dimana memerlukan data- data yang mendukung untuk terlaksananya suatu penelitian.

Penelitian dilakukan menggunakan metode analisis komparatif, Analisis komparatif merupakan salah satu teknik analisis data kuantitatif yang bertujuan untuk mengetahui adanya perbedaan atau tidak pada dua jenis data variabel, membuat generalisasi berdasarkan cara pandang atau pola pikir, menyelidiki hubungan sebab-akibat dengan berdasarkan pengamatan tertentu[11]. Adapun alur penelitian yang dilakukan dapat dilihat pada Gambar 1. 1.



Gambar 1. 1 Alur Penelitian

1.5.1 Identifikasi Masalah

Pada tahap ini, langkah awal yang harus dilakukan adalah melakukan identifikasi masalah yang dihadapi dengan menggunakan arsitektur sistem sebelumnya.

1.5.2 Metode Pengumpulan Data

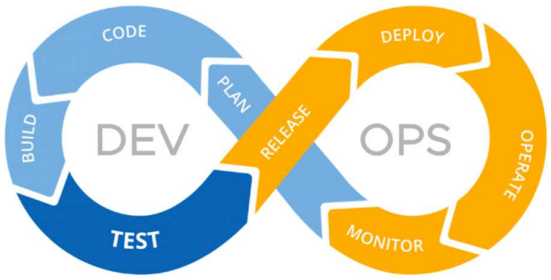
Proses pengumpulan data dalam penelitian ini menggunakan studi literatur dan studi pustaka. Hal ini bertujuan untuk mengumpulkan penelitian terkait yang dapat dijadikan sebagai referensi dalam penelitian yang akan dilakukan.

Tahapan yang dilakukan, dimulai dari pencarian melalui jurnal dan juga e-book. Setelah berbagai referensi terkait terkumpul, kemudian dilanjutkan dengan mencari informasi yang dapat digunakan sebagai landasan teori, metode penelitian, dan segala hal lainnya yang terkait dengan penelitian. Pencarian referensi terkait didasarkan untuk mempelajari beberapa hal sebagai berikut:

1. Penelitian terdahulu mengenai pengimplementasian metode Gitops dibeberapa Kasus
2. Penelitian terdahulu mengenai deployment aplikasi berbasis Cloud Native menggunakan metode push based deployment
3. Penelitian terdahulu mengenai kubernetes di berbagai kasus

1.5.3 Metode Pembangunan Perangkat Lunak

Penelitian ini mengadopsi metode DevOps. Adapun siklus Pengembangan Perangkat Lunak Berdasarkan Paradigma DevOps seperti pada Gambar 1.2.



Gambar 1. 2 Metode Devops

Area Development meliputi tahap *Plan*, *Code*, *Build*, *Test*. Sedangkan area Operation meliputi *Deploy, Operate, Monitor, Release. Area Development* seterusnya disingkat *Dev*, sedangkan area Operation disingkat dengan *Ops*. Jika kedua kata tersebut dipadukan maka menjadi sebuah brand yaitu "*DevOps*".

1. Plan

   Tahap plan meliputi keseluruhan perencanaan dan perancangan aplikasi yang akan dikembangkan. Tahap ini biasanya dipimpin oleh manajer proyek. Semua aturan, persyaratan, dan umpan balik dari pemangku kepentingan, pemilik proyek, dan bahkan pengguna dikumpulkan dan digunakan untuk membuat peta jalan proyek. Alat yang biasa digunakan untuk pelacakan termasuk Asana, Jira, dan ClickUp.
2. Code

   Pada titik ini, pengembang akan mulai menulis kode untuk aplikasi yang sedang dibangun. Ada banyak tools yang digunakan, salah satu yang populer adalah VSCode. Setelah pengembang menulis kode, dia mendorong atau proses pemeriksaan kode ke dalam repositori terpusat, salah satunya adalah Github.
3. Build

   Setelah kode di push ke repositori, build akan selesai. Build disini bermaksud untuk mengubah kode developer menjadi sebuah aplikasi. Biasanya, sebelum membangun, developer lain mengadakan diskusi untuk mendapatkan feedback dan review kode. Setelah pembahasan selesai, langkah selanjutnya adalah membangun aplikasi. Build aplikasi dapat menggunakan berbagai alat bergantung pada aplikasi yang dibuat. Jika file mengonversi ke gambar, proses konversi dapat menggunakan Docker. Atau bisa juga dikonversi menjadi file terkompresi (zip, jar, dll).
4. Test

   Langkah selanjutnya adalah menguji aplikasi yang dibangun. Apakah aplikasi yang dihasilkan memenuhi kriteria, apakah berfungsi dengan baik, sesuai desain, dll. Jika tidak cocok, itu akan berhenti pada titik ini dan

meningkatkan perbaikan. Namun jika cocok, selanjutnya adalah melakukan Rilis.

5. Release

   Selama fase rilis, aplikasi yang lulus fase pengujian diberi label atau versi. Kapan aplikasi dirilis, perubahan apa yang dilakukan, dan kapan aplikasi dirilis sebelum akhirnya di-deploy?

6. Deploy

   Deploy adalah proses penerapan atau penerapan aplikasi yang dibuat dan akhirnya membuatnya dapat diakses oleh pengguna. Salah satu alat yang digunakan adalah AWS CodeDeploy, Jenkins, dll.

7. Operate

   Selama tahap operasi, tim operasi memastikan bahwa aplikasi dan infrastruktur berjalan dengan baik. Juga dapatkan data kinerja, kesalahan, dll. Pengguna juga dapat memberikan umpan balik jika ditemukan kesalahan atau bug. Hal ini akan menjadi benchmark untuk pengembangan aplikasi nantinya.

8. Monitor

   Monitor adalah fase terakhir dari siklus hidup DevOps. Dari fase sebelumnya dikumpulkan dalam bentuk data kinerja, bug atau umpan balik. Data ini dapat digunakan untuk melakukan self-assessment atau evaluasi terhadap aplikasi yang sedang dikembangkan. Fase ini juga dapat memantau alur yang dibuat untuk hambatan di masa mendatang yang dapat memengaruhi produktivitas pengembangan aplikasi tersebut.

1.5.4 Analisis dan Perancangan Sistem

Pada tahap Analisis dan Perancangan untuk implementasi GitOps menggunakan ArgoCD, langkah-langkah kunci mencakup analisis kebutuhan sistem, perancangan arsitektur GitOps, dan pemilihan teknologi yang sesuai. Analisis kebutuhan sistem melibatkan pemahaman mendalam tentang proses pengembangan perangkat lunak, tata kelola kode, serta kebutuhan keamanan dan ketersediaan. Perancangan arsitektur GitOps mencakup identifikasi repositori kode,

lingkungan pengujian, dan produksi, serta konfigurasi ArgoCD untuk otomatisasi pengiriman perangkat lunak. Pemilihan teknologi yang tepat melibatkan evaluasi opsi penyimpanan dan integrasi dengan alat CI/CD yang sesuai dengan kebutuhan organisasi. Tahapan ini menghasilkan desain solusi GitOps yang efektif dengan ArgoCD, yang mendukung otomatisasi pengiriman perangkat lunak secara aman dan andal dalam lingkungan yang dapat diandalkan.

1.5.5 Implementasi dan Pengujian Sistem

Pada tahap implementasi dan pengujian sistem GitOps menggunakan ArgoCD, langkah-langkah kunci mencakup instalasi, konfigurasi perangkat lunak, dan integrasi dengan infrastruktur yang ada. Proses ini melibatkan pemasangan ArgoCD dan penyusunan repositori kode serta definisi aplikasi yang akan dikelola secara otomatis. Selanjutnya, dilakukan integrasi dengan sistem CI/CD yang sudah ada untuk memastikan alur kerja GitOps berjalan lancar.

Setelah implementasi, dilakukan pengujian sistem secara menyeluruh untuk memastikan ketersediaan dan keandalan solusi GitOps dengan ArgoCD. Pengujian mencakup uji beban untuk mengevaluasi kinerja sistem, uji ketahanan untuk memeriksa respons sistem terhadap tekanan eksternal, uji keamanan untuk mengidentifikasi potensi celah keamanan, serta uji pemulihan bencana untuk memastikan sistem dapat pulih dari kegagalan dengan cepat dan efisien. Pengujian dilakukan dengan membandingkan performa dan keandalan sistem yang diimplementasikan menggunakan GitOps dengan ArgoCD dengan arsitektur sebelumnya. Langkah-langkah pengujian ini mencakup evaluasi kinerja, ketersediaan, dan keamanan sistem.

1.5.6 Kesimpulan dan Saran

Pada tahap kesimpulan dan saran, dilakukan evaluasi hasil penelitian yang telah dilakukan dan identifikasi kelebihan dan kekurangan dari solusi konsep gitops yang akan diterapkan. Selain itu, diberikan saran tentang bagaimana solusi konsep gitops dapat ditingkatkan atau dikembangkan lebih lanjut dimasa depan. Tujuannya adalah untuk memberikan gambaran keseluruhan tentang keberhasilan dan

relevansi dari solusi konsep gitops yang telah di terapkan, serta memberikan pandangan untuk pengembangan di masa depan.

1.6 Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran secara umum mengenai permasalahan yang terjadi. Sistematika penulisan penelitian ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Pada Bab ini berisi uraian latar belakang masalah, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, tahap pengumpulan data, model pengembangan perangkat lunak dan sistematika penulisan

BAB 2 TINJAUAN PUSTAKA

Pada Bab ini membahas berbagai konsep-konsep dasar dan teori-teori pendukung yang berhubungan dengan pembangunan sistem.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Pada Bab ini membahas tentang deskripsi sistem, analisis kebutuhan dalam pembangunan sistem serta perancangan sistem.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada Bab ini berisi hasil implementasi analisis dari Bab 3 dan perancangan aplikasi yang dilakukan, serta hasil pengujian aplikasi untuk mengetahui apakah aplikasi yang dibangun sudah memenuhi kebutuhan.

BAB 5 KESIMPULAN DAN SARAN

Pada Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian sistem,serta saran untuk pengembangan.