

MAKALAH

JENKINS DEVOPS TOOLS

Dosen Pengampu : Endang Anggiratih, S.T., M.Cs.



Oleh

Muhammad Hafidz (5200411405)

Ristu Aji Wijayanto (5200411407)

Ikhsan Akbar (5200411435)

Tegar Rangga Nur Ridawan (5200411439)

Metodologi Design Perangkat Lunak Praktik XIV

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

2021

KATA PENGANTAR

Puji syukur kami ucapkan kehadirat Allah SWT atas segala rahmat-Nya sehingga makalah ini dapat tersusun sampai dengan selesai.

Tidak lupa kami mengucapkan terima kasih kepada Ibu Endang selaku Dosen pengampu Mata Kuliah Metodologi Perangkat Lunak Praktik yang telah memberikan ilmu materi guna menyelesaikan tugas mata kuliah ini.

Kami sangat berharap semoga makalah yang kami buat ini dapat menambah pengetahuan dan pengalaman bagi pembaca. Bahkan kami berharap lebih jauh lagi agar makalah ini bisa berguna dalam kehidupan sehari-hari. Bagi kami sebagai penyusun merasa bahwa masih banyak kekurangan dalam penyusunan makalah ini karena keterbatasan pengetahuan dan pengalaman Kami. Untuk itu kami sangat mengharapkan kritik dan saran demi perbaikan makalah ini untuk kedepannya.

Yogyakarta, 9 Desember 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	1
BAB 2	2
PEMBAHASAN	2
2.1 Apa yang dimaksud metode DevOps.....	2
2.2 Apa tujuan metode DevOps	3
2.3 Apa yang dimaksud Jenkins Tools	4
2.4 Apa saja kelebihan dan kekurangan Jenkins Tools.....	4
BAB 3	5
PENGUNAAN TOOLS DEVOPS	5
BAB 4	6
PENUTUP.....	6

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam suatu proses perancangan sebuah aplikasi tentunya dibutuhkan suatu metode yang tepat dan tentunya agar membantu ketika dalam proses perancangan dan pengembangan aplikasi. Ini sangat penting karena akan menentukan hasil akhir dari sebuah aplikasi namun kerap kali dalam melakukan proses menemui masalah dikarenakan sulitnya dalam menentukan metode yang tepat dan cepat dalam pengembangan aplikasi. Dan dari masalah tersebut kami memilih metode DevOps yang merupakan bagian dari Agile Development sebagai solusi dimana dibutuhkan metode yang cepat dan tepat dalam menangani berbagai request client.

DevOps adalah metode pengembangan *software* yang menekankan pada komunikasi, kolaborasi dan integrasi antara *developer* dan professional TI (Avila & Kunaidi, 2021). DevOps merupakan definisi dari *Development* dan *Operation* yang memiliki prinsip developer untuk mengkoordinasikan antar tim *Development* dengan tim *Operations* dengan efektif dan efisien sehingga bisa dilakukan dengan singkat dan tidak membutuhkan banyak pertanyaan. Dalam pengembangan menggunakan metode DevOps tentunya juga membutuhkan sebuah *tools*. Banyak *tools* yang bisa digunakan, salah satunya adalah Jenkins yang biasa digunakan secara umum atau *open-source* oleh semua tim *development*. Jenkins sendiri merupakan *tools* yang bersifat *open-source* yang bisa digunakan untuk berbagai integrasi yang terkait dengan pembuatan, pengujian dan penerapan *software*.

1.2 Rumusan Masalah

1. Apa yang dimaksud metode DevOps
2. Apa tujuan metode DevOps
3. Apa yang dimaksud Jenkins Tools
4. Apa saja kelebihan dan kekurangan Jenkins Tools

1.3 Tujuan

1. Mengetahui apa yang dimaksud Metode DevOps
2. Mengetahui tujuan dari metode DevOps
3. Mengetahui apa yang dimaksud Jenkins Tools
4. Mengetahui kelebihan dan kekurangan Jenkins Tools

BAB 2

PEMBAHASAN

2.1 Apa yang dimaksud metode DevOps

DevOps adalah seperangkat prosedur yang merupakan kombinasi antara proses pengembangan (development) dan operasi (operations). DevOps juga dapat diartikan sebagai studi konseptual tentang pengembangan dan pengiriman perangkat lunak ke dalam infrastruktur dengan mengambil suatu pendekatan kolaboratif dan integrative antara pengembang (develop) dan operasi perangkat lunak (operation).

DevOps membutuhkan seperangkat alat untuk menunjukkan fungsi dari kombinasi dan integrasi. Dengan kata lain DevOps adalah tim tunggal yang menangani pengembangan, pengujian, dan operasi di siklus produk total yang tidak pernah putus

DevOps juga dapat diartikan sebagai seperangkat praktik dan alat yang didasarkan pada perangkat lunak dan rekayasa system. Siklus hidup DevOps Metode DevOps memiliki siklus hidup dalam pengembangan proyek perangkat lunak, urutan siklus hidup yang ada pada metode DevOps

- **Plan**

Pada tahap perencanaan, proses identifikasi tujuan dan persyaratan untuk merancang dan mengembangkan perangkat lunak, selain itu kegiatan lain yang dilakukan pada tahapan ini yaitu manajemen proyek, penjadwalan, rencana perilis, kebijakan/persyaratan, serta rencana awal untuk pembaharuan dan perilis di seluruh literasi

- **Develop**

Dari tahapan sebelumnya, tim pengembang berfokus untuk mengembangkan dan meninjau kode perangkat lunak atau IaC. Kode aplikasi dikembangkan dengan menggunakan Bahasa pemrograman dan IDE yang sesuai dan dikelola dengan menggunakan system control versi

- **Build**
Pada tahap ini integrasi dari berbagai modul perangkat lunak dilakukan untuk membuat file yang dapat dieksekusi untuk fitur produk atau produk yang dikembangkan sepenuhnya
- **Test**
Pada tahap ini pengujian otomatis dilakukan secara terus menerus untuk memastikan kualitas perangkat lunak yang telah dikembangkan.
- **Deploy**
Tahap ini berfokus pada penerapan ulang perangkat lunak secara terus menerus di lingkungan produksi. Fase ini melibatkan masalah manajemen konfigurasi platform dan sumber daya target.
- **Operate**
Operasi dalam siklus DevOps berkaitan dengan konfigurasi dan pengelolaan aplikasi perangkat lunak setelah penerapan, misalnya penyediaan sumber daya dan penskalaan otomatis.
- **Monitor**
Kinerja aplikasi yang diterapkan dipantau dengan mengumpulkan dan menganalisis data penggunaan yang dapat membantu mendeteksi dan mengidentifikasi pengecualian dan memberikan umpan balik untuk peningkatan perangkat lunak secara berulang

2.2 Apa tujuan metode DevOps

DevOps bertujuan untuk meningkatkan kolaborasi antara tim development dan tim operation dari mulai perencanaan hingga aplikasi/ fitur ter-deliver ke pengguna, semua itu harus dilakukan secara otomatis agar :

- Meningkatkan deployment frequency
- Mengkatkan waktu pemasaran
- Menurunkan tingkat kegagalan pada rilis terbaru
- Mempersingkat waktu perbaikan
- Meningkatkan waktu pemulihan

2.3 Apa yang dimaksud Jenkins Tools

Jenkins merupakan sebuah solusi open source yang menyediakan Continuous Integration dan Continuous Delivery secara otomatis dan sebagai tools pengujian otomatis untuk aplikasi Cloud Native, dengan Jenkins penggunaan tidak perlu mengeluarkan banyak usaha dan waktu dalam menkonfigurasi ekosistem yang kompleks, Jenkins melakukan Sebagian besar beban pekerjaan tersebut

Jenkins mengotomatiskan CI/CD untuk Cloud dan mencari tahu cara mendapatkan konfigurasi, plug in, dan kode yang tepat agar bisa bekerja sama, Jenkins memberikan dukungan lengkap untuk CD dan mengelola produk di environment preview, produksi, dan staging

Singkatnya Jenkins merupakan tools DevOps dalam membantu pengguna mengurangi rasa khawatir tentang konfigurasi infrastruktur yang mendasar, seperti instalasi dan konfigurasi, serta akan mendapatkan lebih banyak waktu untuk pengembangan actual dan mempercepat alur kerja CI /CD

2.4 Apa saja kelebihan dan kekurangan Jenkins Tools

A. Kelebihan Jenkins

- Mudah untuk di setup
- Mempromosikan environment baru melalui GitOps
- Recovery yang cepat
- Isolasi
- Kecepatan
- Rilis lebih cepat

B. Kekurangan Jenkins

- Terbatas pada Kubernetes. Aspek deploy Jenkins berkelanjutan ditunjukkan untuk namespace kubernetes
- Memerlukan Akses Level ke Level Cluster-Admin Kubernetes
- Jenkins terbatas pada proyek Git

BAB 3

PENGUNAAN TOOLS DEVOPS

Judul	Implementasi CI/CD Pipeline Pada Framework Androbase Menggunakan Jenkins (Studi Kasus: PT. Andromedia)
Jurnal	Nasional Komputasi dan Teknologi Informasi
Volume & Halaman	Vol 4 No 6
Tahun	Desember 2021
Penulis	Ahmad Farid ,Indra Gita Anugrah
Reviewer	Ristu Aji Wijayanto (5200411407)
Tanggal	11 Januari 2022
Tujuan Penelitian	Mencari solusi dari permasalahan PT Andromedia saat proses deployment aplikasi
Metode Penelitian	<p>Dalam metode penelitian ini memiliki tahapan-tahapan yang dibuat dalam acuan dalam pembuatan penelitian. Tahapan pertama adalah identifikasi masalah, pada tahapan ini penulis mengidentifikasi masalah pada development aplikasi di PT. Andromedia dimana pada saat development aplikasi masih menggunakan cara manual.</p> <p>Tahap kedua adalah rumusan masalah dimana pada tahapan ini penulis mengajukan solusi dengan menggunakan CI/CD</p> <p>Tahapan selanjutnya adalah implementasi CI/CD dengan menggunakan Jenkins. Untuk Tahapan terakhir adalah pengujian dan hasil kesimpulan dimana aplikasi yang telah terintegrasi dengan Jenkins akan di uji dan dari pengujian itu dapat ditarik kesimpulan</p>
Hasil Penelitian	<p>Penelitian ini diawali dengan melakukan instalasi git pada computer masing-masing developer dan jenkins pada server lokal, setelah aplikasi yang di butuhkan terinstall maka langkah selanjutnya adalah membuat repository baru di github kemudian framework androbase di masukan (push) ke repository tersebut.</p> <p>Untuk langkah berikutnya adalah membuat job pipeline pada jenkins yang kemudian job tersebut diintegrasikan dengan</p>

	<p>repository github yang telah dibuat, Setelah job terintegrasi dengan github maka langkah selanjutnya adalah mendefinisikan stage stage yang akan di eksekusi apabila terdapat perubahan pada repository github stage stage yang dibuat</p> <p>Kesimpulannya:</p> <p>Dari hasil penelitian dapat disimpulkan bahwa dengan menerapkan sistem CI/CD pipeline dengan menggunakan aplikasi jenkins dapat mempermudah proses deploymen aplikasi dimana proses deployment yang awalnya menggunakan cara manual dengan CI/CD dapat dilakukan secara otomatis cara ini juga dapat menghemat waktu dan juga dapat meminimalisir kesalahan. Dengan menggunakan sistem ini semoga dapat membantu developer PT.Andromedia dalam mendevelop aplikasi.</p>
--	--

BAB 4

PENUTUP

PERBANDINGAN MODEL PENGEMBANGAN PERANGKAT LUNAK

Tahapan Pengembangan Perangkat Lunak	Waterfall	Prototype	RAD	countinuous integration (Agile)	DevOps
Perencanaan Sistem (Systems Planning)	Berawal dari kebutuhan	Berawal dari kebutuhan	Berawal dari kebutuhan	Berawal dari kebutuhan	Berawal dari kebutuhan
Analisis Sistem (Systems Analysis)	<p>Kebutuhan data harus dianalisis diawal secara lengkap dan menyeluruh</p> <p>Perubahan data ataupun fungsional akan merubah keseluruhan proses pada tahapan berikutnya.</p>	<p>Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing.</p> <p>Perubahan dapat dilakukan selama sistem atau perangkat lunak masih dalam bentuk prototype</p>	<p>Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing</p> <p>Kebutuhan fungsi mayor dapat dimodulkan dalam waktu tertentu dan dapat dibicarakan oleh tim RAD yang terpisah.</p>	<p>Kebutuhan data harus dianalisis diawal secara lengkap dan menyeluruh</p> <p>Kebutuhan fungsional dan data dapat dimodulkan dalam waktu tertentu dan dapat dibicarakan oleh tim Continuous Integration</p>	<p>Kebutuhan data harus dianalisis diawal secara lengkap dan menyeluruh</p> <p>Kebutuhan fungsional dan data dapat dimodulkan dalam waktu tertentu dan dapat dibicarakan oleh tim DevOps</p>
Perancangan Sistem	Testing dilakukan ketika semua tahapan pada model sudah selesai.	Testing dapat dilakukan ketika prototype telah dibangun, sehingga hasil testing dapat merubah rancangan sistem.	<p>Testing dapat dilakukan ketika prototype telah dibangun, sehingga hasil testing dapat merubah rancangan sistem.</p> <p>Memberikan prototype</p>	Testing dilakukan pada saat tahapan integration	Testing dilakukan secara terus menerus saat tahapan pengembangan

(Systems Design)	Tidak dapat memberikan gambaran yang jelas mengenai sistem yang dibangun, karena sistem bisa dilihat jika semua tahapan telah dilakukan.	Memberikan prototype sebagai gambaran sistem yang akan dibangun, sehingga user dapat melihat dan berinteraksi langsung dengan gambaran sistem. User berperan aktif dalam pengembangan system Sistem yang dibangun akan sesuai dengan keinginan user	sebagai gambaran system yang akan dibangun, sehingga user dapat melihat dan berinteraksi langsung dengan gambaran sistem. User berperan aktif dalam pengembangan system Sistem yang dibangun akan sesuai dengan keinginan user Mempunyai kemampuan untuk menggunakan kembali komponen yang ada (reusable object) sehingga pengembang tidak perlu membuat dari awal lagi dan waktu lebih singkat	dapat memberikan gambaran yang jelas mengenai sistem yang dibangun, User berperan aktif dalam pengembangan system Sistem yang dibangun akan sesuai dengan keinginan user	dapat memberikan gambaran yang jelas mengenai sistem yang dibangun, User berperan aktif dalam pengembangan system Sistem yang dibangun akan sesuai dengan keinginan user
	Menerapkan proses perancangan yang baik	Tidak menerapkan proses perancangan yang baik	Tidak menerapkan proses perancangan yang baik	Menerapkan proses perancangan yang baik	Menerapkan proses perancangan yang baik

Implementasi Sistem (Systems Implement)	Evaluasi dilakukan setelah system telah dibangun	Evaluasi dilakukan ketika prototype telah dibangun	Evaluasi dilakukan ketika prototype telah dibangun	Evaluasi dilakukan Ketika Integration	Evaluasi dilakukan Ketika Pengembangan berlangsung
	Mengedepankan kebutuhan fungsional sistem	Mengedepankan aspek kenyamanan user	Mengedepankan aspek kenyamanan user dan kecepatan pembangunan	Mengedepankan aspek kenyamanan user dan kecepatan pembangunan	Mengedepankan aspek kenyamanan user dan kecepatan pembangunan
Pemeliharaan Sistem (Systems Maintenance)	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan

Hasil yang dicapai berdasarkan penelitian dan perbandingan yang telah dilakukan diatas, maka dapat disimpulkan:

- 1) Dapat diketahui karakteristik dari kelima model pengembangan perangkat lunak Waterfall, Prototype, RAD, Continous Integration dan DevOps
- 2) Model pengembangan Waterfall cocok digunakan untuk sistem atau perangkat lunak yang bersifat generik, artinya sistem dapat diidentifikasi semua kebutuhannya dari awal dengan spesifikasi yang umum serta sesuai untuk perangkat lunak yang memiliki tujuan untuk membangun sebuah sistem dari awal yang mengumpulkan kebutuhan sistem yang akan dibangun sesuai dengan topik penelitian yang dipilih sampai dengan produk tersebut diuji
- 3) Model pengembangan Prototype lebih cocok untuk sistem atau perangkat lunak yang bersifat customize, artinya software yang diciptakan berdasarkan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai untuk perangkat lunak memiliki tujuan untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus.
- 4) Model pengembangan RAD lebih cocok untuk sistem atau perangkat lunak yang bersifat customize, berskala besar dan memerlukan waktu yang lebih singkat artinya software yang diciptakan berdasarkan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai untuk perangkat lunak memiliki tujuan untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus, serta memiliki kemungkinan untuk kebutuhan pengembangan kembali dalam jangka waktu yang cukup panjang.
- 5) Model pengembangan Continous Integration hampir sama seperti RAD cocok untuk sistem atau perangkat lunak yang bersifat customize, berskala Medium dan Proses pengembangannya mendorong kolaborasi, organisasi, dan umpan balik sebagai sarana untuk menyelesaikan proyek lebih cepat dan lebih efisien.

6) Model pengembangan DevOps hampir sama seperti RAD dan CI cocok untuk sistem atau perangkat lunak yang bersifat customize, berskala Besar dan Proses kolaboratif antara tim pengembangan dan operasi IT yang digunakan sepanjang siklus pengembangan perangkat lunak yang membutuhkan waktu relatif cepat

7) Metode-metode yang dianalisa mempunyai kelebihan dan kekurangannya masing-masing sehingga tidak dapat ditentukan mana yang lebih baik. Dari kelebihan dan kekurangan masing-masing metode, pengembang dapat memilih metode mana yang paling cocok untuk dirinya.