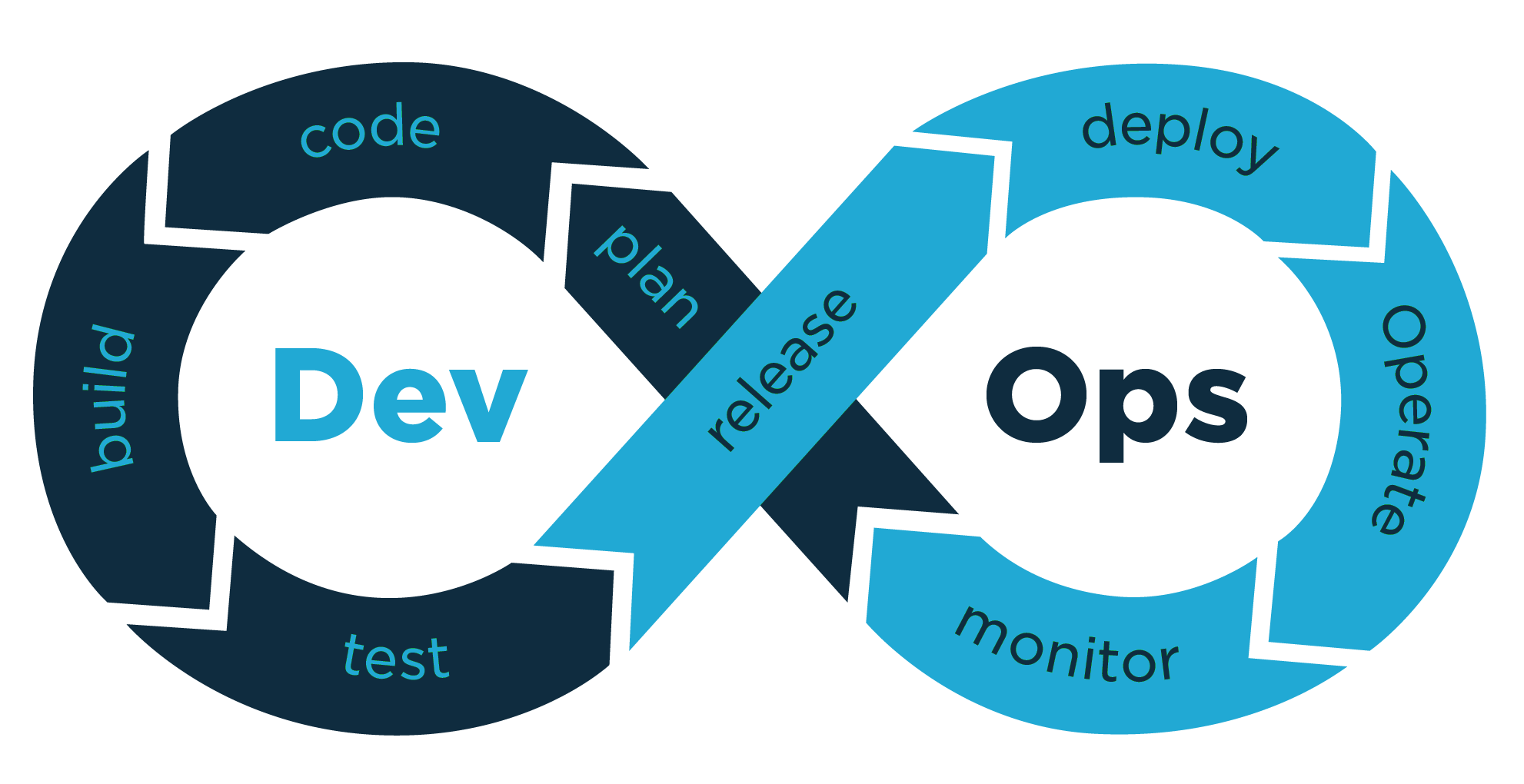
Mengadopsi DevOps di PT Meteor Inovasi Digital

Sebuah perusahaan PT Meteor Inovasi Digital mengadopsi sebuah DevOps yang dipimpin oleh seorang leader yang bernama David. DevOps singkatan dari Development and Operations yang merupakan sebuah budaya atau culture baru untuk mengkoordinasikan antar tim yaitu tim development dengan tim operations dengan efektif dan efisien.

Eno adalah seorang DevOps Engineer yang akan bergabung dalam tim DevOps dan ditugaskan untuk memberikan penjelasan mengenai adopsi DevOps di perusahaan PT Meteor Inovasi Digital.

1. Pengertian DevOps



DevOps adalah kombinasi dari culture, penerapan, dan peralatan yang meningkatkan kemampuan perusahaan atau organisasi untuk memberikan aplikasi dan layanan dalam kecepatan yang tinggi, mengembangkan dan meningkatkan produk dalam waktu yang lebih cepat dari organisasi yang menggunakan pengembangan software dan proses pengelolaan infrastruktur tradisional. Kecepatan ini membuat organisasi dapat melayani customer dengan lebih baik dan dapat berkomptensi dengan lebih efektif di pasaran.

Koordinasi yang diterapkan pada sebuah DevOps membutuhkan sebuah tools. Ada banyak tools yang bisa digunakan salah satunya adalah Source Code Management (SCM) yang biasa digunakan secara umum oleh tim development. Produk SCM yang paling terkenal adalah Git, ditemani oleh Source Code Repository (SCR) seperti GitHub, GitLab, Bitbucket, atau yang lainnya. Namun SCM saja tak cukup untuk mengomunikasikan antara tim development dengan tim operational.

Untuk mengetahui permasalahan yang terjadi, biasanya tim operational akan dihubungkan ke Product Management Software seperti Jira. Melalui Produk Management Software, tim operation dapat mengetahui berbagai permasalahan yang terjadi pada sistem/aplikasi. Sehingga antara pihak development dengan operational akan saling terhubung satu sama lain.

1. Tujuan DevOps

DevOps bertujuan untuk mempercepat pengiriman produk software yang lebih baik dan berkualitas tinggi serta meningkatkan kolaborasi antara tim development dan tim operation, mulai dari perencanaan hingga aplikasi/fitur ter-deliver ke user. Semua itu harus dilakukan secara otomatis agar:

1. Meningkatkan deployment frequency.
2. Meningkatkan waktu pemasaran.
3. Menurunkan tingkat kegagalan pada rilisan terbaru.
4. Mempersingkat waktu perbaikan.
5. Meningkatkan waktu pemulihan.
6. Manfaat DevOps

Manfaat DevOps yang dapat dirasakan oleh sebuah organisasi adalah sebagai berikut

1. Kecepatan

Bergerak dalam kecepatan tinggi sehingga perusahaan dapat melakukan inovasi untuk pelanggan dengan lebih cepat, beradaptasi dengan lebih baik di pasar yang berubah, dan menjadi lebih efisien dalam mendorong hasil bisnis yang berkualitas tinggi. Model DevOps memungkinkan tim pengembang dan pengoperasian perusahaan untuk mencapai hasil tersebut. Misalnya, [layanan mikcro](https://aws.amazon.com/id/devops/what-is-devops/" \l "microservices) dan [pengiriman berkelanjutan](https://aws.amazon.com/id/devops/continuous-delivery/) memungkinkan tim mendapatkan kepemilikan layanan dan kemudian meluncurkan pembaruannya dengan lebih cepat.

1. Pengiriman yang cepat

Dapat meningkatkan frekuensi dan kecepatan peluncuran sehingga perusahaan dapat berinovasi dan meningkatkan produk dengan lebih cepat. Semakin cepat dapat meluncurkan fitur baru dan memperbaiki bug, semakin cepat pula dapat merespons kebutuhan pelanggan dan mendapatkan keuntungan yang kompetitif. [Integrasi berkelanjutan](https://aws.amazon.com/id/devops/continuous-integration/) dan [pengiriman berkelanjutan](https://aws.amazon.com/id/devops/continuous-delivery/) merupakan praktik yang mengautomasi proses peluncuran software, dari pembuatan hingga penerapan.

1. Keandalan

Keandalan yang terjadi seperti integrasi berkelanjutan dan pengiriman berkelanjutan untuk menguji bahwa tiap perubahan berfungsi aman. Praktik pemantauan dan pembuatan log dapat membantu mendapatkan informasi performa secara waktu-nyata.

1. Skala

Automasi dan konsistensi membantu Anda mengelola sistem yang kompleks atau berubah secara efisien dan dengan risiko yang berkurang. Misalnya, [infrastruktur sebagai kode](https://aws.amazon.com/id/devops/what-is-devops/" \l "iac) membantu Anda mengelola pengembangan, pengujian, dan lingkungan produksi dengan cara yang dapat diulang dan lebih efisien.

1. Kolaborasi yang dapat ditingkatkan

Tingginya tingkat kolaborasi antara tim development dan tim operation dapat mengurangi kesalahan dan meningkatkan kualitas produk.

1. Keamanan

Keamanan yang ada pada DevOps dapat menentukan dan melacak kegiatan pada skala yang telah ditetapkan. Sehingga model DevOps dapat digunakan tanpa mengorbankan keamanan dengan menggunakan kebijakan kepatuhan yang terautomasi, control yang menyeluruh dan teknik manajemen konfigurasi.

1. Tahapan-Tahapan Siklus DevOps

Siklus DevOps mencakup serangkaian kombinasi yang secarra umum dikenal sebagai “DevOps Lifecycle”. Tahapan-tahapannya yaitu sebagai berikut:

1. Plan

Tahap ini melibatkan perencanaan proses pengembangan dan pengiriman software. Tim DevOps merencanakan apa yag harus dilakukan selama siklus ini dan membuat rencana untuk memenuhi tujuan bisnis. Tujuan utamanya adalah untuk memastikan bahwa semua anggota tim memahami tujuan, ruang lingkup, dan batasan proyek atau produk.

Tools yang digunakan Trello, JIRA, Asana atau GitLab (untuk manajemen tugas dan proyek) dan Google Docs atau Microsoft Office (untuk kolaborasi dokumen).

1. Code

Tahap ini melibatkan pengembangan dan penulisan code yang akan digunakan untuk membangun aplikasi. Tim DevOps biasanya menggunakan alat seperti Git atau SVN untuk mengelola kode sumber dan berkolaborasi pada kode yang sama. Kode harus dipelihara dengan baik agar mudah dikelola di kemudian hari.

Tools yang digunakan yaitu Git, GitHub, atau GitLab (untuk manajemen kode sumber) dan Visual Studio Code, IntelliJ IDEA atau Sublime Text (untuk text editor pengembangan code).

1. Build

Tahap ini melibatkan membangun dan menguji kode yang telah ditulis dan dikembangkan. Tujuannya adalah untuk memastikan bahwa kode dapat dijalankan dan berfungsi sesuai dengan kebutuhan. Hal ini dapat dilakukan secara otomatis dengan menggunakan tools seperti Jenkins atau Travis CI, GitLab CI/CD, CircleCI

1. Test

Tahap ini melibatkan pengujian kode atau aplikasi yang telah dibangun untuk memastikan bahwa ia berfungsi dengan baik dan sesuai dengan kebutuhan. Hal ini juga untuk memastikan kualitas produk yang dihasilkan serta pengujian keamanan dan kinerja. Tim DevOps biasanya melakukan pengujian otomatis dan manual untuk memastikan aplikasi berfungsi dengan baik.

Tool yang digunakan Selenium, Appium atau TestComplete (untuk pengujian fungsional), JMeter atau LoadRunner (untuk pengujian beban), dan SonarQube (untuk pengujian keamanan).

1. Release

Tahap ini melibatkan persiapan untuk merilis aplikasi ke lingkungan produksi. Tim DevOps akan memeriksa bahwa aplikasi telah diuji dengan baik dan sudah siap untuk di release. Ini termasuk dokumentasi, pelaporan, dan persiapan infrastruktur.

Tools yang digunakan yaitu Jenkins, CircleCI, atau Gitlab CI/CD (untuk integrasi berkelanjutan) dan Docker atau Kubernetes (untuk manajemen kontainer).

1. Deploy

Tahap ini melibatkan pengiriman aplikasi ke lingkungan produksi. Tim DevOps akan menggunakan alat seperti Ansible atau puppet untuk mengotomasi proses pengiriman produk.

Tools yang digunakan Ansible atau Chef (untuk manajemen konfigurasi), Terraform atau CloudFormation (untuk manajemen infrastruktur) dan AWS CodeDeploy atau Google Cloud Deployment Manager (untuk pengiriman code).

1. Operate

Tahap ini melibatkan operasi aplikasi dalam lingkungan produksi. Tim DevOps memastikan bahwa aplikasi berjalan dengan baik dan dapat diakses oleh pengguna.

Tools yang digunakan yaitu Nagios, Zabbix atau Promotheus (untuk monitoring kinerja).

1. Monitor

Tahap ini melibatkan pemantauan aplikasi dalam lingkungan produksi. Tim DevOps akan memantau performa aplikasi dengan melibatkan pengumpulan dan analisis data untuk memantau kinerja produk yang telah dirilis.

Tools yang digunakan ELK Stack atau Splunk (untuk analisis log).

Setiap alat yang dipilih tergantung pada kebutuhan tim development dan tim operation serta lingkungan teknis yang digunakan.

1. Cara kerja DevOps di Industri IT

Berikut ini adalah beberapa langkah umum dalam cara kerja DevOps di industry IT:

1. Integrasi berkelanjutan : tim development dan operation bekerja sama untuk melakukan integrasi berkelanjutan yaitu membangun, menguji, dan merilis software secara otomatis dan terus menerus.
2. Pengujian otomatis : DevOps juga memanfaatkan pengujian otomatis untuk memastikan bahwa software yang dihasilkan sesuai dengan kebutuhan user.
3. Provisioning otomatis : DevOps juga memanfaatkan provisioning otomatis untuk mempercepat pemasangan dan konfigurasi infrastruktur IT
4. Monitoring dan analisis : DevOps melibatkan monitoring dan analisis infrastruktur IT dan aplikasi untuk memastikan bahwa mereka berjalan dengan baik dan memberikan pengalaman yang baik bagi pengguna.
5. Kode yang dikelola dengan baik : DevOps juga mendorong tim pengembangan untuk mengelola kode dengan baik sehingga mudah untuk dikerjakan, dikembangkan dan dipelihara.
6. Kolaborasi yang lebih baik : DevOps meningkatkan kolaborasi antara tim development dan operation, sehingga memungkinkan mereka untuk lebih cepat merespon perubahan pasar dan kebutuhan user.

Dengan mengadopsi DevOps, perusahaan dapat meningkatkan efisiensi, mempercepat waktu rilis produk, meningktakan kualitas produk dan meminimalkan risiko kesalahan ynag terjadi selama proses pengembangan dan operasi.