

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Agate adalah perusahaan pengembang *video game* Indonesia yang berbasis di Bandung, Jawa Barat, Indonesia, didirikan pada 1 April 2009 [1]. Agate memulai perjalanan sebagai studio pengembangan *game*, Agate mulai memberanikan diri untuk terjun dalam dunia bisnis pada tahun 2010 dengan memanfaatkan gamifikasi untuk meningkatkan *engagement* dan motivasi di tempat kerja. Produk yang dihasilkan oleh Agate dalam bidang bisnis ini berfokus pada produk yang bisa meningkatkan sistem pembelajaran, pengembangan diri, sistem penilaian dan kegiatan daring. Agate telah melayani lebih dari 200 klien di berbagai macam bidang industri [2]. Pada tahun 2022 Agate menggunakan logo pada gambar 1-1.



Gambar 1-1 logo resmi Agate

Sumber: glints.com

Dalam menjalankan bisnisnya Agate menggunakan dua jenis produk untuk menghasilkan keuntungan yaitu *Consumer Product* dan *Enterprise Business*. Adapun *Consumer Product* merupakan jenis produk yang berfokus pada *player*, umumnya merupakan sebuah game yang diperuntukan untuk bersenang-senang. Sedangkan pada *Enterprise Business* berfokus pada korporasi besar yang membutuhkan gamifikasi untuk meningkatkan penjualan, loyalitas, pengembangan dan penilaian karyawan. Umumnya gamifikasi yang digunakan digunakan sebagai *Marketing Solutions* atau *Learning Solutions*.

Oleh sebab itu Agate seringkali berurusan dengan berbagai macam proyek. Banyak dari proyek tersebut membutuhkan *server* untuk menjalankan *backend* untuk produknya. Hal ini membuat proses pembuatan server sering dilakukan dan rawan terjadi *human error*. Maka dari itu otomatisasi menjadi hal yang penting untuk proses pembuatan *server* agar kemungkinan terjadinya *human error* dapat dihindari. Selain itu proses otomatisasi juga dapat memberikan manfaat karena prosesnya lebih *efficient* dimana biaya produksi menjadi lebih sedikit.

1.2 Rumusan Masalah

Berdasarkan observasi yang telah dilakukan selama magang, Agate membutuhkan otomatisasi konfigurasi server menggunakan Ansible.

1.3 Tujuan

Tujuan dari dibuatnya proyek akhir ini yaitu untuk menerapkan proses otomatisasi konfigurasi server menggunakan Ansible guna meminimalisir terjadinya *human error*.

1.4 Batasan Masalah

Batasan masalah dapat berisi:

1. *Tools* yang digunakan adalah Ansible dan AWX,
2. AWX dijalankan pada Kubernetes,
3. AWX diinstal menggunakan Kubernetes Operator,
4. Banyaknya modul yang dibuat sesuai dengan kebutuhan Agate,
5. Modul dibuat berdasarkan SOP Agate.

1.5 Definisi Operasional

Ansible sering digambarkan sebagai *configuration management tool* dan biasanya disandingkan dengan Puppet, Chef dan Salt. Saat seorang IT professional membicarakan tentang *configuration management*, yang dimaksudkan adalah menuliskan kondisi yang diinginkan terhadap sebuah server. Lalu menggunakan sebuah *tools* untuk memastikan server telah terkonfigurasi sesuai dengan yang diharapkan mulai dari package yang terinstall, perijinan akun dan layanan yang berjalan pada server. Ansible adalah salah satu *tools* yang digunakan untuk hal tersebut.

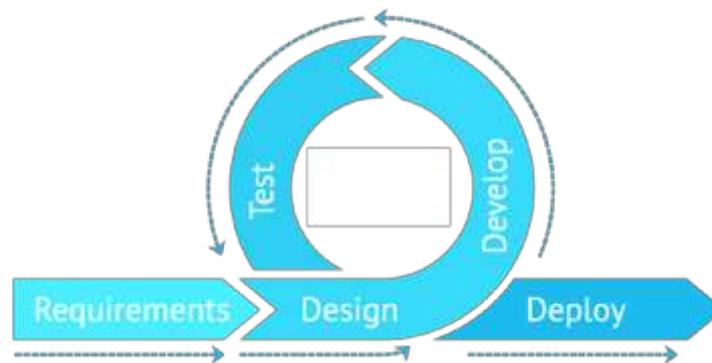
AWX adalah sebuah *Web Interface open-source* yang digunakan untuk mengoperasikan Ansible. AWX merupakan alternative dari Ansible Tower yang merupakan sebuah layanan berbayar. AWX cocok digunakan pada fase *development* dan Ansible Tower cocok digunakan pada fase *production*. Keduanya pada dasarnya adalah sebuah aplikasi yang sama, hanya saja Ansible Tower memiliki stabilitas yang lebih baik.

Automation adalah sistem atau teknologi yang dapat mengelola sesuatu secara otomatis.

IaC merupakan singkatan dari *Infrastructure as Code*, adalah merupakan sebutan untuk *management infrastructure* yang menggunakan *text files* (code) dalam mendefinisikan konfigurasinya. *Text files* lebih mudah untuk diduplikasi dan di customisasi daripada sistem GUI. Oleh karena itu, IaC sering digunakan dalam proses Automatisasi.

1.6 Metode Pengerjaan

Pada proses pengerjaan proyek akhir ini metode yang digunakan adalah metode *Agile* yang sedikit dimodifikasi untuk mempercepat proses *testing* dan *release*. Metode *Agile* dipilih karena dapat mempercepat proses pembuatan modul, *testing* dan hasil modulnya dapat digunakan dalam waktu yang lebih cepat. Dengan metode ini pencarian masalah dan solusi dapat dilakukan lebih awal.



Gambar 1-2 Metode pengerjaan

Tahapan metode *Agile* yang sedikit dimodifikasi dibagi menjadi 5

1.6.1 Requirement

Tahap *requirement* adalah tahap untuk mengumpulkan kebutuhan-kebutuhan apa saja yang perlu dicapai pada suatu *Playbook*. Di tahap ini penting sekali untuk berkonsultasi dengan atasan agar hasil dari *Playbook* yang dibuat sesuai dengan kebutuhan perusahaan. Di tahap ini juga atasan umumnya akan memberikan *link* ke dokumentasi SOP yang sudah dimiliki oleh perusahaan.

1.6.2 Design & Develop

Pada tahap ini proses pembuatan modul mulai dilakukan

1.6.3 Test

Pada tahap ini dilakukan proses pengujian untuk mengukur sejauh mana sebuah modul telah menyelesaikan kebutuhan tanpa terjadi masalah apapun. Apabila terjadi masalah maka alur prosesnya akan kembali ke *Design & Develop* hingga semua masalah telah terselesaikan.

1.6.4 Deploy

Secara umum tahap ini tidak memerlukan perlakuan khusus karena semua modul ditulis pada branch master, selanjutnya selain pembuat modul dapat menggunakan modul yang dibuat langsung via AWP. Selain itu dokumentasi dilakukan setelah semua proses telah selesai.