

BAB I

PENDAHULUAN

Bab pendahuluan ini membahas mengenai latar belakang masalah, rumusan masalah, tujuan dan manfaat, ruang lingkup, serta sistematika penulisan yang akan digunakan dalam penelitian yang berjudul Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*.

1.1. Latar Belakang Masalah

Docker merupakan sebuah *platform* yang digunakan untuk mengembangkan, mengirimkan, dan menjalankan aplikasi dalam *container*. *Container* merupakan suatu teknologi virtualisasi yang dapat digunakan untuk mengemas aplikasi dan semua dependensinya sehingga dapat didistribusikan dan dijalankan pada infrastruktur lainnya (Yadav dkk., 2019).

Docker didistribusikan dengan *command-line client* yang dapat dijalankan pada *terminal* dan sebuah API yang digunakan untuk berinteraksi dengan *background process* yaitu *daemon* yang digunakan untuk mengelola seluruh komponen *Docker* (Turnbull, 2018). Bagi pengguna awam yang baru belajar *Docker*, penggunaan *command-line* untuk mengelola *Docker* tidak efektif karena harus menghafal sintaks dari setiap perintah yang ada. Alternatif yang dapat digunakan adalah aplikasi yang berbasis *graphical user interface* (GUI) karena pengguna cukup dengan menekan tombol dan memasukkan input pada form yang tersedia untuk mengelola *Docker*. Oleh karena itu, pengembangan aplikasi GUI yang digunakan untuk mengelola *Docker* dapat dilakukan dengan manfaat memberikan alternatif yang lebih mudah bagi pengguna awam yang ingin belajar mengenai *Docker*.

Dalam pengembangan perangkat lunak terdapat beberapa macam metode pengembangan perangkat lunak yang dapat digunakan salah satunya adalah *ICONIX Process*. Rosenberg & Stephens (2007) menjelaskan bahwa *ICONIX Process* merupakan suatu metode pengembangan perangkat lunak yang menitikberatkan pada tahap antara pembuatan *use case* dan penulisan kode. Menurut Ristyawan & Harini (2019), pendekatan *ICONIX Process* didasarkan pada notasi *Unified Modeling*

Language (UML) seperti yang terdapat pada *Rational Unified Process* (RUP) akan tetapi tidak serumit seperti yang diimplementasikan pada RUP. Lingkup pengembangan *ICONIX Process* sempit dan singkat seperti *Extreme Programming* (XP) tetapi masih menggunakan tahap analisis dan desain. *ICONIX Process* berada di tengah-tengah antara RUP dan XP, penggunaan UML yang lebih efisien sehingga implementasi menjadi lebih mudah.

Platform yang digunakan untuk aplikasi yang dikembangkan adalah berbasis *web* sehingga aplikasi dapat dijalankan pada sistem operasi apapun menggunakan *web browser* selama *Docker* sudah didukung oleh sistem operasi tersebut.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang dijelaskan pada sub bab sebelumnya dapat dirumuskan suatu permasalahan yaitu bagaimana mengembangkan aplikasi GUI berbasis *web* untuk manajemen *Docker* menggunakan metode *ICONIX Process*.

1.3. Tujuan dan Manfaat

Tujuan dari dilaksanakannya tugas akhir ini adalah menghasilkan aplikasi GUI berbasis *web* untuk manajemen *Docker* menggunakan metode *ICONIX Process*.

Manfaat yang ingin dicapai dari dikembangkannya aplikasi pada tugas akhir ini adalah pengguna awam yang ingin belajar *Docker* dapat menggunakan alternatif dari *command-line* untuk mengelola *Docker* melalui sebuah aplikasi GUI yang dioperasikan cukup dengan menekan tombol dan memasukkan input yang tersedia pada tampilan antarmuka sehingga proses belajar menggunakan *Docker* lebih efektif.

1.4. Ruang Lingkup

Ruang lingkup dalam Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*:

1. Aplikasi yang dikembangkan merupakan aplikasi berbasis *web*.
2. Aplikasi hanya mencakup beberapa fitur yang terdapat pada API *Docker* untuk mengelola *container* yaitu menampilkan daftar *container*, membuat *container*

baru, menghapus *container*, memperbarui data *container*, dan mengatur proses *container*.

3. Aplikasi hanya mencakup beberapa fitur yang terdapat pada API *Docker* untuk mengelola *image* yaitu menampilkan daftar *image*, membuat *image* baru melalui *pull image*, membuat *image* baru melalui *build image*, menghapus *image*, dan memperbarui versi *image* dengan *push image* ke Docker Hub.
4. Aplikasi hanya mencakup beberapa fitur yang terdapat pada API *Docker* untuk mengelola *volume* yaitu menampilkan daftar *volume*, membuat *volume* baru, menghapus *volume*.
5. Aplikasi hanya mencakup beberapa fitur yang terdapat pada API *Docker* untuk mengelola *network* yaitu menampilkan daftar *network*, membuat *network* baru, menghapus *network*, dan menghubungkan *network* dengan *container*.
6. Aplikasi hanya berfokus sebagai alat bantu pengembangan perangkat lunak dalam skala kecil atau satu individu, sehingga aplikasi tidak perlu menggunakan autentikasi.

1.5. Sistematika Penulisan

Sub bab ini memberikan gambaran bagian-bagian yang menyusun laporan atau sistematika penulisan laporan yang digunakan untuk Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*.

BAB I PENDAHULUAN

Bab ini membahas latar belakang masalah, rumusan masalah, tujuan dan manfaat, ruang lingkup, serta sistematika penulisan laporan Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*.

BAB II LANDASAN TEORI

Bab ini berisi studi literatur dan landasan teori yang digunakan sebagai acuan untuk menyelesaikan permasalahan yang terdapat pada Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*.

BAB III METODOLOGI PENELITIAN

Bab ini menyajikan informasi mengenai metodologi penelitian yang digunakan untuk menyelesaikan permasalahan yang terdapat pada Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil implementasi penelitian dan pembahasan terkait Pengembangan Aplikasi Manajemen *Docker* berbasis *Web* menggunakan metode *ICONIX Process*.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hal yang sudah dipaparkan pada seluruh bab sebelumnya serta saran untuk memperbaiki pengembangan selanjutnya.