

BAB 1

USULAN GAGASAN

1.1 Latar Belakang Masalah

Perkembangan teknologi membuat peluang bisnis khususnya pada bidang teknologi informasi terbuka lebar sehingga muncul banyak perusahaan teknologi [1]. Perusahaan yang berbasis teknologi informasi biasanya menyediakan layanan-layanan seperti infrastruktur dan aplikasi kepada klien. Pelanggan terkadang mengalami masalah pada bagian infrastruktur atau aplikasi sehingga klien perlu melaporkan masalah tersebut ke perusahaan penyedia layanan infrastruktur dan aplikasi. Masalah yang sering terjadi yaitu keterlambatan penanganan masalah karena akses pelaporan klien kepada vendor tidak dapat diandalkan. Perusahaan tidak memiliki sistem yang dapat diandalkan untuk mengelola pelaporan klien secara terstruktur untuk memberikan tanggapan dan solusi yang tepat sesuai dengan kebutuhan dan masalah yang dialami pelanggan [2].

Sistem yang dibuat harus menerapkan arsitektur perancangan perangkat lunak yang efektif dan reliabel sehingga meningkatkan kepuasan pelanggan terhadap layanan yang diberikan. Solusi dari permasalahan tersebut yaitu penggunaan aplikasi *ticketing system* dimana aplikasi ini berfungsi sebagai tempat untuk melaporkan setiap permasalahan yang dihadapi [3]. Permasalahan yang telah selesai akan secara otomatis terdokumentasi dan dapat dijadikan sebagai referensi jika terdapat masalah yang sama [4]. Aplikasi yang digunakan saat ini adalah aplikasi dengan arsitektur monolitik yang menyatukan seluruh komponen ke dalam satu sistem. Arsitektur tersebut digunakan untuk memudahkan pembuatan aplikasi karena fitur dan komponen aplikasi disatukan dalam satu unit atau satu servis yang utuh.

Seiring berkembangnya aplikasi dengan penambahan fitur maka aplikasi menjadi besar dan menyebabkan lambatnya proses pengembangan fitur baru. Semua komponen terikat pada satu unit atau satu servis yang sama sehingga perubahan kecil bisa memiliki dampak yang luas pada seluruh aplikasi. Permasalahan tersebut dapat diselesaikan dengan menerapkan arsitektur *microservice*. Arsitektur *microservices* memisahkan komponen menjadi beberapa servis yang dapat berdiri sendiri oleh karena itu servis tidak akan mengganggu servis lainnya sehingga pengembang dapat lebih mudah melakukan pengembangan fitur baru [5]. Capstone ini menggunakan rancangan *Microservices* untuk pengembangan aplikasi tiket helpdesk yang

diharapkan bisa menjawab kebutuhan akan ketersediaan yang tinggi dan juga kemudahan dalam pengembangan kedepannya [6].

Berdasarkan keterangan dari perusahaan terkait, permasalahan yang dihadapi dalam sistem ticketing saat ini adalah keterbatasan pengembangan karena menggunakan aplikasi pihak ketiga. Penggunaan aplikasi tersebut menyulitkan perusahaan karena harus melaporkan ke vendor terkait terkait bug atau permintaan pengembangan. Project capstone ini akan merancang platform sesuai kebutuhan perusahaan akan lebih memudahkan dan mengefisienkan kebutuhan dan operasional. Aplikasi ticketing yang dibuat akan mengusung konsep *microservices* yang diharapkan bisa menjawab kebutuhan akan ketersediaan yang tinggi [6]. Monitoring dan logging diperlukan untuk melakukan pemantauan terhadap aplikasi dan infrastruktur sehingga memastikan sistem berjalan dengan baik. Monitoring dan logging dilakukan untuk mendeteksi apabila terjadi anomali pada sistem dan membantu untuk mencari akar permasalahan apabila terjadi error ataupun anomali dalam sistem yang berjalan [1].

1.2 Informasi Pendukung Masalah

Perusahaan sering kali mendapatkan kendala teknis dengan sistem ticketing saat ini. Sistem yang berjalan saat ini menggunakan aplikasi pihak ketiga dimana aplikasi ini sering kali mengalami malfungsi. Pihak ketiga merupakan vendor source code management terintegrasi sehingga sulit untuk menyesuaikan fitur dan kebutuhan perusahaan. Aplikasi yang didesain dan dibuat dengan fitur yang dapat dikustomisasi diharapkan dapat mengatasi masalah-masalah tersebut.

1.3 Analisis Umum

1.3.1 Aspek Manajemen

Divisi support di perusahaan Boer Technology terdiri dari beberapa orang teknisi yang bertugas untuk mengatasi permasalahan terkait infrastruktur dari sebuah aplikasi yang dimiliki oleh pelanggan. Apabila pelanggan mengalami masalah pada layanan komputasi awan yang diberikan perusahaan Boer Technology, maka client akan melapor ke teknisi melalui aplikasi pesan singkat seperti telegram atau whatsapp. Pelanggan akan melapor permasalahannya ke teknisi secara detail setelah itu teknisi langsung membuat tiket di aplikasi pihak ketiga. Permasalahan yang tertulis dalam tiket akan ditangani oleh teknisi yang sedang bertugas. Aplikasi pihak ketiga ini memiliki beberapa kekurangan yang menyulitkan perusahaan dalam memberikan pelayanan kepada pelanggan.

Kekurangan yang pertama yaitu pada aplikasi pihak ketiga ini tidak terdapat informasi tentang estimasi penyelesaian suatu tiket. Informasi tersebut penting karena sering terjadi kesalahpahaman antara teknisi dan pelanggan dalam waktu penyelesaian masalah atau tiket. Kekurangan yang kedua yaitu pelanggan tidak memiliki akses ke platform ticketing dari pihak ketiga sehingga pelanggan harus menghubungi engineer secara pribadi. Engineer yang mendapatkan laporan seharusnya langsung membuat tiket namun dalam beberapa kasus engineer lupa untuk membuat tiket di aplikasi pihak ketiga. Kasus tersebut membuat masalah yang dialami pelanggan tidak tercatat dan tidak dapat ditangani dengan baik oleh engineer.

Kekurangan yang ketiga yaitu jika terdapat bug atau masalah pada aplikasi pihak ketiga maka harus melaporkan masalah tersebut ke vendor. Ketergantungan terhadap vendor membuat penggunaan sistem ticketing tidak maksimal karena vendor tersebut lambat dalam menanggapi masalah yang dilaporkan oleh perusahaan. Kekurangan yang keempat yaitu untuk meminta fitur baru sesuai kebutuhan perusahaan sangatlah sulit karena ini merupakan aplikasi dari pihak ketiga. Aplikasi ticketing yang dirancang sendiri dan memiliki fitur-fitur yang dapat dikustomisasi akan menjadi solusi untuk memajemen permasalahan yang dialami pelanggan.

1.3.2 Aspek Kontrol

Aplikasi pihak ketiga yang digunakan oleh divisi support di perusahaan Boer Technology ini mempunyai kekurangan dalam aspek kontrol. Aspek tersebut berhubungan erat dengan kontrol perusahaan terhadap teknisi yang menangani permasalahan pelanggan secara langsung. Perusahaan tidak mengetahui bagaimana *feedback* pelanggan terhadap performa teknisi yang menyelesaikan suatu masalah. Kurangnya *feedback* serta sistem pemantauan yang tidak efektif membuat manajer atau pimpinan tidak bisa melihat bagaimana penilaian pelanggan terhadap performa para teknisi. Fitur yang dapat dikustomisasi dapat menyelesaikan permasalahan aspek kontrol dengan membuat fitur *feedback* serta analisis performa engineer [2].

1.4 Kebutuhan yang Harus Dipenuhi

Kebutuhan yang harus dipenuhi untuk menyelesaikan permasalahan tersebut adalah sebuah platform ticketing system untuk divisi support. Platform tersebut berfungsi untuk mengelola sebuah laporan masalah dan request, dimana laporan masalah dan request ini disebut tiket. Kebutuhan teknis dari perusahaan dan fungsinya sebagai platform ticketing harus terpenuhi dengan beberapa fitur yang dapat dikustomisasi.

Platform yang dirancang mempunyai beberapa fitur :

- A. Membuat tiket dengan beberapa keterangan
- B. Komentar pada tiket
- C. Memberikan tiket ke seorang teknisi
- D. Menutup tiket setelah permasalahan terselesaikan
- E. Membuat tiket otomatis ketika pelanggan mengirim email ke salah satu akun
- F. Estimasi penyelesaian tiket
- G. Pengelompokan klien dan user
- H. Feedback dari pelanggan kepada teknisi
- I. Pemantauan performa engineer
- J. Analisis data berkaitan tiket, engineer, serta klien
- K. Berjalan diatas lingkungan *Cloud Native (Orkestrasi Kontainer)* beserta stack *self hosted* pendukung jalannya *SDLC* aplikasi *microservices*

1.5 Solusi Sistem yang Diusulkan

Kami menawarkan beberapa solusi untuk menyelesaikan masalah laporan yang diberikan oleh pelanggan layanan komputasi awan. Solusi yang kami tawarkan yaitu aplikasi ticketing system berbasis *microservice* dan aplikasi ticketing system berbasis monolitik.

1.5.1 Karakteristik Produk

Produk harus bisa memecahkan masalah sebelumnya, khususnya pada hal ketersediaan servis dan dapat diandalkan untuk proses *tracking* performa engineer maupun client.

1.5.1.1 Aplikasi Ticketing System dengan Arsitektur *Microservices*

Aplikasi ticketing system memiliki fitur untuk melakukan pendaftaran akun berdasarkan role yang tersedia dimana setiap user harus melewati proses autentikasi [7]. Pelanggan dapat membuat laporan mengenai keluhan dan masalah dari layanan yang diberikan oleh perusahaan. Masalah yang dilaporkan oleh klien dapat langsung eskalasi oleh teknisi sehingga masalah dapat diselesaikan dengan segera. Aplikasi menggunakan arsitektur *microservice* yang terdiri dari beberapa layanan kecil yang dapat berkomunikasi di layer aplikasi menggunakan protokol *Representational State Transfer (REST)* dan menggunakan *Application Programming Interface (API)* dimana semua data akan dikirim dalam format *JSON*. Layanan backend dapat berkomunikasi dengan sub bagian backend dan data akan ditampilkan oleh layanan frontend yang digunakan langsung oleh klien [8].

Aplikasi layanan mikro yang dibuat membutuhkan runtime dimana layanan tersebut berjalan. layanan tersebut membutuhkan server komputasi awan yang berjalan di atas virtualisasi level container. Antar container dapat berhubungan satu sama lain dan berjalan diatas container engine seperti containerd. Objek virtualisasi container membutuhkan sebuah platform untuk melakukan orkestrasi terhadap container dan membutuhkan overlay network serta lingkungan yang dapat menjalankan aplikasi tersebut. Cluster container orchestration dipengaruhi oleh spesifikasi mesin server komputer yang digunakan, semakin tinggi spesifikasi maka kinerja semakin baik. Fitur utama pada level infrastruktur adalah penyembuhan otomatis, penyeimbangan beban, mengatur trafik menggunakan konsep software defined network, hingga monitoring .

Ketersediaan dan performansi merupakan faktor penting dalam penggunaan suatu aplikasi. Karakteristik dari aplikasi layanan mikro yaitu ketika satu layanan berhenti berjalan maka layanan lainnya tidak akan terpengaruh karena setiap layanan dapat berjalan sendiri atau bersifat independen, hal tersebut juga dapat mempermudah pengembangan aplikasi karena kode program setiap service tidak bergantung dengan kode program servis lainnya. Penerapan aplikasi yang berbasis microservices ini diharapkan dapat memudahkan pengembang sistem untuk memudahkan *scaling* layanan tertentu dan memudahkan menambahkan fitur baru pada aplikasi.

1.5.1.2 Aplikasi Ticketing System Berbasis Monolitik

Fitur pada solusi ini mempunyai kesamaan dengan solusi yang pertama namun arsitekturnya berbeda, pada solusi yang kedua ini kami menggunakan arsitektur monolitik. Arsitektur monolitik adalah suatu model arsitektur aplikasi di mana seluruh fitur dan komponen aplikasi disatukan dalam satu unit atau kode sumber yang utuh. Karakteristik dari arsitektur monolitik ini yaitu rancangan dari aplikasi yang berarsitektur monolith terdiri dari 1 codebase atau servis yang berisi semua fitur sehingga semua komponen terikat pada satu unit yang sama. Aplikasi yang berbasis monolitik lebih mudah dibuat dan dipantau karena semua komponen terikat pada satu unit yang sama. Penerapan aplikasi yang berbasis monolitik ini diharapkan dapat mempermudah pembuatan aplikasi.

1.5.2 Skenario Penggunaan

Solusi platform yang diajukan memiliki beberapa role untuk memudahkan proses penggunaan platform. Tabel 1.1 menunjukkan role dalam platform ticketing serta peruntukan role tersebut. Dalam implementasi nyata, client akan membuat laporan melalui email atau

dashboard website yang dibuat. Otomatis setelah tiket terbuat baik admin maupun engineer dapat melihat di masing masing dashboard sehingga memudahkan semua pihak untuk tracking perkembangan status laporan.

Tabel 1.1 Skenario Penggunaan

Role	Pemakaian
Admin	ditujukan untuk para direksi perusahaan cloud services
Engineer	ditujukan untuk team leader dan engineer pada perusahaan cloud service.
Client	ditujukan untuk client dari perusahaan cloud service

1.5.2.1 Aplikasi Ticketing System dengan Arsitektur Microservices

Fitur atau komponen dari aplikasi ticketing system berbasis microservices ini dibuat menjadi beberapa servis yang terpisah, dimana setiap servis tersebut mempunyai fungsi yang berbeda. Aplikasi ticketing system berbasis microservices ini terdiri dari frontend service, comment service, member service, ticket service, email service, analytic service, dan performance service. Frontend service berfungsi menangani frontend dari aplikasi, saat pengguna mengakses aplikasi, maka pengguna akan langsung menuju ke frontend service. Member service berfungsi untuk menangani hal terkait akun seperti login, membuat akun, dan menghapus akun. Ticket service berfungsi menangani hal yang berkaitan dengan tiket seperti pembuatan tiket, dan penghapusan tiket. Comment service berfungsi untuk menangani hal yang terkait dengan komentar pada tiket seperti pembuatan komentar pada tiket.

Email service berfungsi untuk menangani hal terkait penerimaan email, email service ini berintegrasi dengan ticket service untuk pembuatan tiket dari sebuah email, saat pengguna atau klien mengirimkan sebuah email yang berisi judul keluhan dan isi keluhan ke email sistem, maka email service ini akan memproses email tersebut dan berintegrasi dengan ticket service untuk membuat tiket dari email yang dikirimkan. Analytic service berfungsi untuk menghimpun data-data berkaitan dengan platform ticketing seperti jumlah tiket berdasarkan status (open, closed, in progress). Performance service berfungsi untuk melakukan pemberian feedback atau penilaian untuk teknisi, menampilkan data performa dari teknisi seperti nilai performa, dan menampilkan waktu rata-rata pengerjaan tiket yang dikerjakan oleh teknisi.

1.5.2.2 Aplikasi Ticketing System Berbasis Monolitik

Fitur atau komponen dari aplikasi ticketing system berbasis microservices ini dibuat menjadi satu kesatuan dalam satu unit atau kode sumber yang utuh. Dimana nantinya front end, fitur tiket, fitur comment, fitur member, fitur analytic dan fitur email akan dibuat pada 1 codebase. Sehingga nantinya arsitektur dari aplikasi ticketing system akan kelihatan lebih sederhana dan mudah dipahami dan dipelihara. Masalah yang dihadapi jika memilih arsitektur monolitik yaitu ketika aplikasi bertambah besar dan membutuhkan fitur baru akan sulit untuk mengembangkannya dan aplikasi akan menjadi berat seiring bertambahnya kompleksitas aplikasi.