BAB 1
PENDAHULUAN

1.1 Latar Belakang

Pemanfaatan teknologi komputasi awan memungkinkan perusahaan untuk mengatur skalabilitas sumber daya IT sesuai dengan kebutuhan bisnis mereka. Ini berarti bahwa saat beban kerja meningkat, perusahaan dapat dengan cepat meningkatkan kapasitas komputasi dan penyimpanan mereka. Sebaliknya, jika beban kerja menurun, mereka dapat menurunkan sumber daya yang digunakan. Ini memungkinkan respons yang cepat terhadap perubahan dalam permintaan bisnis, yang sangat penting dalam lingkungan yang berubah dengan cepat[1].

Kubernetes merupakan platform *open source* sistem orkestrasi kontainer untuk mengotomatisasi *deployment*, *scaling*, dan manajemen aplikasi. Aplikasi yang sudah di kontainerisasikan dapat dijalankan di OS dan komputasi awan mana pun[2]. Platform ini menggunakan *Docker* sebagai dasar untuk menjalankan portabel kontainer atau disebut dengan *Docker images*. Platform ini juga menyediakan fitur *Horizontal Pod Autoscaler* (HPA) yang mampu menyeimbangkan dan mengantisipasi beban kerja yang tinggi pada *Docker* dengan menyesuaikan beban kerja yang berlangsung[3].

Jabar Digital Service (JDS) adalah Unit Pelaksana Teknis Daerah (UPTD) yang berada di bawah Dinas Komunikasi dan Informatika Provinsi Jawa Barat. JDS merupakan tim yang menyediakan aplikasi untuk pemesanan tiket *west java festival* pada 2-3 September 2023.

Permasalahan yang muncul pada saat pemesanan tiket dibuka adalah banyaknya pengguna yang mengakses aplikasi pada waktu yang sama untuk mendapatkan tiket, karena terlalu tingginya beban kerja terhadap server, beberapa pengguna gagal mendapatkan tiket. Proses autoscaling yang terjadi bahkan beberapa kali gagal dan membuat server down sejenak. Dari data yang didapatkan terjadi kasus permasalahan yang terjadi berulang kali, ketika *response latency* mencapai lebih dari 50ms akan terjadi kegagalan pada request selanjutnya. Aplikasi pemesanan

tiket *west java festival* sudah menerapkan *kubernetes* dengan menggunakan *Kubernetes Resource Metrics* (KRM).

Pada penelitian ini penulis berfokus untuk menganalisa KRM dengan kustom metrik ingress nginx untuk mencari solusi dalam mengatasi permasalahan yang terjadi, lebih spesifiknya membandingkan metriks CPU dan Memory pada KRM dengan *response latency* pada ingress nginx maupun keduanya. Sehingga analisis ini diharapkan dapat memberikan hasil data perbandingan metrik untuk menemukan metrik yang efektif dalam penerapan HPA pada pembangunan aplikasi skala besar.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka dapat dirumuskan masalah-masalah yang ada adalah sebagai berikut :

1) Bagaimana implementasi HPA untuk metrik KRM dan *ingress nginx* dalam menjaga skalabilitas sebuah aplikasi dengan *kubernetes* ?
2) Bagaimana pengujian metrik HPA yang dapat memberikan hasil untuk digunakan dalam pembuatan aplikasi ?
3) Bagaimana hasil pengujian metrik HPA dapat memudahkan stakeholder dalam monitoring server ?

1.3 Maksud & Tujuan

Berdasarkan uraian rumusan masalah diatas, maka maksud dari penelitian ini adalah menganalisis metrik KRM (CPU dan Memory) dengan *ingress nginx* (*response latency*) maupun kombinasi keduanya pada HPA, untuk menjaga skalabilitas sebuah aplikasi dengan *kubernetes*. Sedangkan tujuan dilaksanakannya penelitian ini adalah sebagai berikut:

1. Implementasi HPA untuk metrik KRM dan *ingress nginx* dalam menjaga skalabilitas sebuah aplikasi.
2. Memberikan hasil pengujian yang dapat memberikan hasil untuk digunakan dalam pembuatan aplikasi.

3. Memberikan data hasil pengujian yang dapat memudahkan stakeholder dalam memudahkan monitoring terkait metrik HPA.

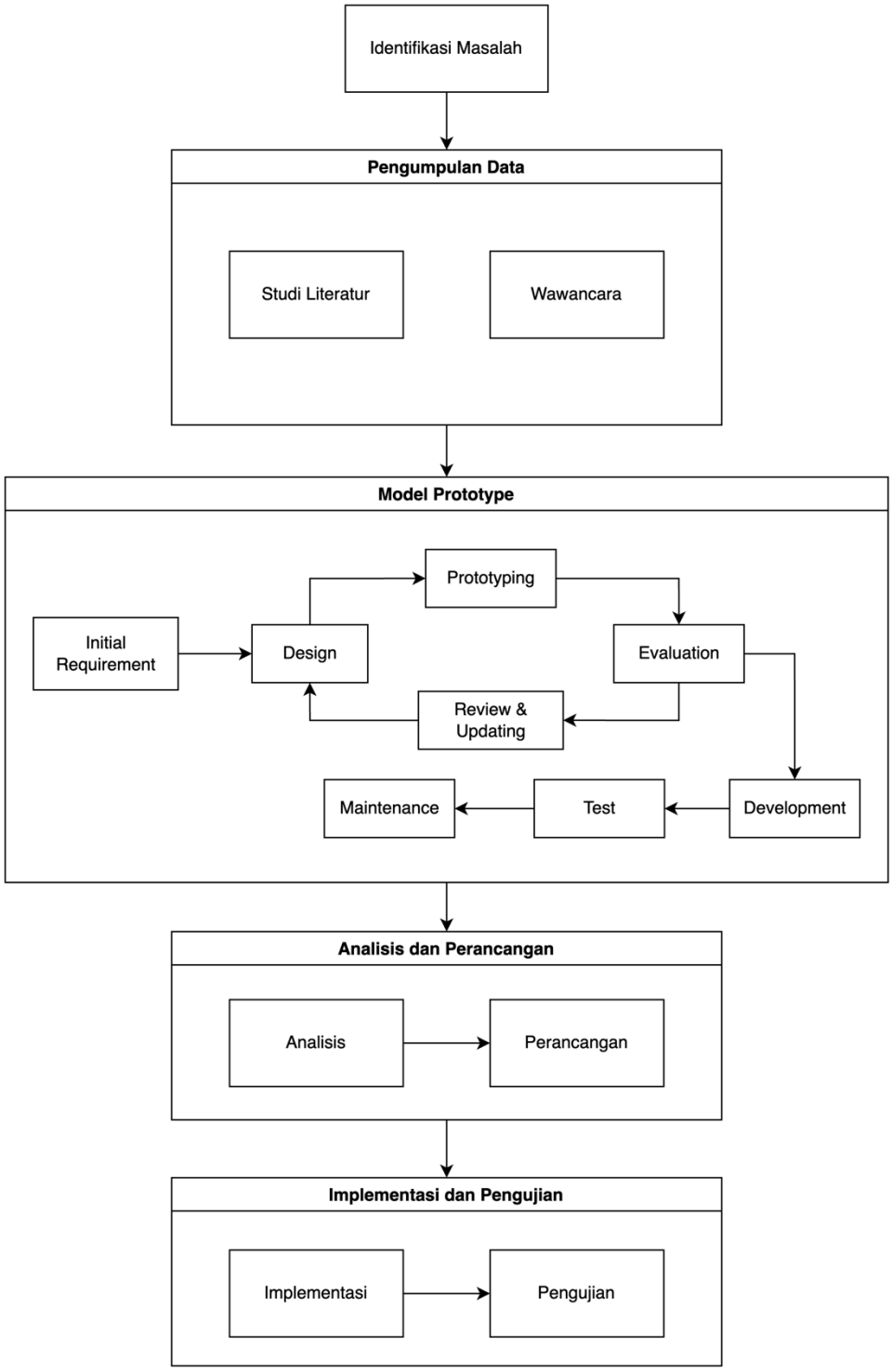
1.4 Batasan Masalah

Adapun batasan-batasan masalah dalam penelitian ini adalah sebagai berikut:

1) Penelitian hanya berfokus pada analisis performansi success rate aplikasi dengan metrik CPU dan Memory, *response latency* dan keduanya.
2) Pengujian pada penelitian ini terbatas pada lingkungan *Kubernetes* dengan jumlah tiga *pods*.
3) *Docker* untuk kontainerisasi aplikasi.
4) Hasil dari penelitian berupa rancangan sistem dan *source code* konfigurasi.

1.5 Metodologi penelitian

Pada penelitian kali ini penulis akan menggunakan metode analisis Komparatif. Analisis komparatif merupakan salah satu teknik analisis data kuantitatif yang bertujuan untuk mengetahui adanya perbedaan atau tidak pada dua jenis data variabel, membuat generalisasi berdasarkan cara pandang atau pola pikir, menyelidiki hubungan sebab-akibat dengan berdasarkan pengamatan tertentu[4].



Gambar 1.1 Gambar Alur Penelitian

1.5.1 Identifikasi Masalah

Pada tahap ini, langkah awal yang harus dilakukan adalah melakukan identifikasi masalah yang dihadapi dengan menggunakan arsitektur sistem sebelumnya.

1.5.2 Metode Pengumpulan Data

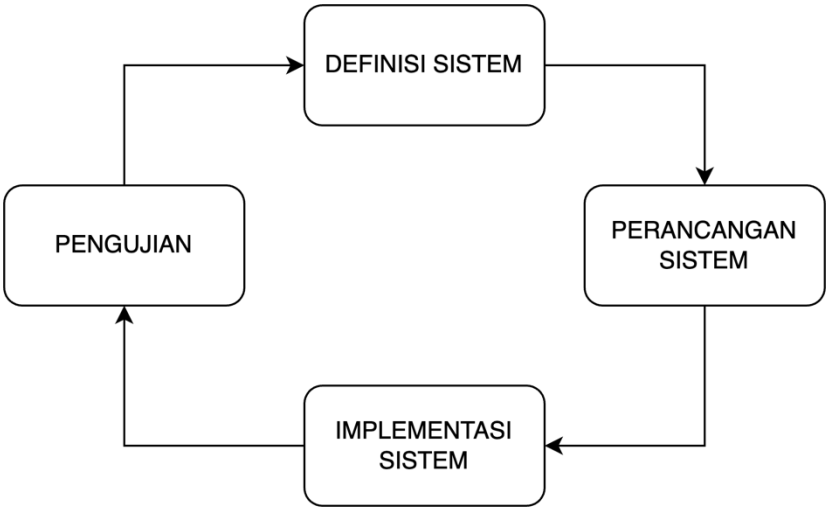
Metode pengumpulan data yang digunakan adalah dengan melakukan studi literatur dan metode wawancara.

Studi literatur dilakukan dengan cara mempelajari, meneliti dan menelaah berbagai literatur-literatur dari perpustakaan yang bersumber dari buku-buku, teks, jurnal dan bacaan-bacaan lainnya yang ada kaitannya dengan topik penelitian.

Wawancara dilakukan kepada tim *devops* JDS, ini bertujuan untuk mengumpulkan data terkait jumlah *request* pengguna saat pemesanan tiket *West Java Festival*. Data *request* pengguna yang diperoleh dari wawancara ini akan digunakan sebagai acuan dalam melakukan *load testing* dalam proses analisis metrik pada *kubernetes*.

1.5.3 Metode Pembangunan Sistem

Metode pembangunan perangkat lunak yang digunakan adalah metode *prototype*. Metode ini dipilih karena hasil dari pembangunan dapat di evaluasi kembali jika terjadi kesalahan. Alur dari metode *prototype* dapat dilihat pada Gambar 1.2



Gambar 1.2 Metode Prototype

1. Definisi Sistem

Pada bagian ini didefinisikan sistem yang akan dibangun yaitu sistem *autoscaling* pada *Kubernetes* dengan menggunakan *Horizontal Pod Autoscaling*.

2. Perancangan Sistem

Perancangan Sistem mencakup konfigurasi metrik apa saja yang akan dimonitor, kapan skala otomatis harus dilakukan dan berapa jumlah replika pod.

3. Implementasi Sistem

Pada bagian ini dilakukan implementasi HPA pada *kubernetes* dengan konfigurasi yang sudah dirancang sebelumnya dan dilakukan *deployment* sebuah aplikasi.

4. Pengujian

Pengujian dilakukan dengan beberapa skenario, seperti memberikan beban tinggi dan rendah ke aplikasi yang sudah dikonfigurasi HPA, lalu melihat perilaku skalabilitasnya, apakah jumlah pod sudah otomatis dinaik

turunkan sesuai konfigurasi atau belum. Pengujian juga dilakukan dengan beberapa metrik yang berbeda, hasil pengujian dianalisis untuk melihat efektivitas konfigurasi *Horizontal Pod Autoscaling* yang dibuat.

1.5.4 Analisis dan Perancangan Sistem

Pada tahap Analisis dan Perancangan Sistem untuk implementasi HPA diperlukan konfigurasi pada arsitektur, beberapa langkah-langkah yang perlu dilakukan antara lain menganalisis kebutuhan spesifikasi sistem seperti CPU dan *memory* untuk menjalankan aplikasi *prototype*. Merancang arsitektur *Kubernetes* seperti *node* dan konfigurasi *port* jaringan. Konfigurasi HPA seperti metrik KRM dan Kustom Metrik *Ingress* dan juga *threshold* yang diperlukan untuk pemicu skala otomatis, dan juga mengatur batas pod replika pada sistem. Merancang konfigurasi *deployment* dan pod untuk aplikasi *prototype* termasuk konfigurasi untuk replika dan sumber daya. Dan juga merancang skenario untuk *load testing* aplikasi *prototype* untuk mengetahui performa hasil dari HPA

1.5.5 Implementasi dan Pengujian

Pada tahap ini dilakukan proses implementasi dari hasil Analisis dan Perancangan Sistem, dan proses pengujian dari hasil implementasi, berikut tahapan-tahapan yang dilakukan :

1. Membuat dan Konfigurasi Kubernetes Cluster dengan Google Cloud Platform (GCP) sesuai dengan rancangan yang telah dibuat.
2. Instalasi dan Konfigurasi Ingress Nginx untuk *load balancer* dan kustom metrik
3. Instalasi dan Konfigurasi Grafana dan Prometheus untuk monitoring
4. Instalasi dan Konfigurasi Prometheus Adapter untuk mendapatkan data kustom metrik
5. Mengimplementasikan konfigurasi *HPA* sesuai dengan rancangan yang telah dibuat.
6. Mengimplementasikan *deployment* aplikasi *prototype* dengan *Kubernetes*

7. Melakukan *Load Testing* sesuai dengan rancangan skenario yang telah dibuat.
8. Mengumpulkan dan menganalisis data yang didapat dari hasil *load testing* untuk membandingkan performa dalam mencari penggunaan metrik yang lebih baik.

1.5.6 Kesimpulan dan Saran

Pada tahap kesimpulan dan saran, dilakukan evaluasi hasil penelitian yang telah dilakukan dan identifikasi kelebihan dan kekurangan dari metrik KRM dan kustom metrik *ingress* yang dibangun. Menilai bagaimana kemampuan skala otomatis HPA dengan metrik yang berbeda. Tujuannya adalah untuk memberikan gambaran tentang keberhasilan dan relevansi dalam menjaga skalabilitas sebuah aplikasi dengan metrik yang berbeda menggunakan HPA *Kubernetes*.

1.6 Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran secara umum mengenai permasalahan dan pemecahannya. Sistematika penulisan skripsi ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Pada Bab ini membahas tentang latar belakang, identifikasi masalah, batasan masalah, tujuan dan manfaat serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Pada Bab ini membahas tentang landasan teori yang berguna untuk penelitian kedepannya.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Pada Bab ini membahas tentang rancangan penelitian, tahapan penelitian, objek penelitian, sistem yang digunakan, teknik pengumpulan dan analisis data, jadwal pelaksanaan, dan rincian biaya.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Pada Bab ini membahas implementasi konfigurasi, pengujian dan analisis hasil dari pengujian yang dilakukan.

BAB 5 KESIMPULAN DAN SARAN

Pada Bab ini berisi simpulan dan saran dari pengujian yang penulis lalukan