

BAB 2

TINJAUAN PUSTAKA

2.1 State Of the Art

Penelitian sebelumnya berfungsi untuk analisa dan memperkaya pembahasan penelitian, serta membedakannya dengan penelitian yang sedang dilakukan. Dalam penelitian ini disertakan lima jurnal internasional penelitian sebelumnya yang berhubungan dengan konsep *brand image*. Jurnal tersebut antara lain pada Table 2.1:

Table 2.1 State Of the Art

No.	Judul Jurnal dan Peneliti	Tahun dan Tempat Penelitian	Metode Penelitian	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
1.	<i>Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration</i>	2020, Korea	kuantitatif	Penggunaan <i>Horizontal Pod Autoscaling</i> (HPA) di <i>Kubernetes</i> untuk mengelola jumlah pod secara elastis berdasarkan beban kerja.	Peneliti membandingkan performa antara KRM dan <i>Prometheus</i> kustom metrik CPU untuk mencari metrik yang baik digunakan dalam pembangunan aplikasi.
2	<i>Adaptive scaling of Kubernetes pods</i>	2020, Hungaria	kuantitatif	Penerapan <i>Libra</i> untuk untuk adaptif skala otomatis, kemampuan <i>libra</i> dalam mengontrol metode HPA dan <i>Vertical Pods Autoscaler</i> (VPA) dalam	Peneliti menggunakan <i>tools</i> <i>Libra</i> untuk membangun skala otomatis yang adaptif, <i>tools</i> ini membangun untuk membuat skala otomatis tanpa perlu adanya

No.	Judul Jurnal dan Peneliti	Tahun dan Tempat Penelitian	Metode Penelitian	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
				proses skala otomatis	konfigurasi secara konvensional.
3	<i>Building Modern Clouds: Using Docker, Kubernetes & Google Cloud Platform</i>	2019, Amerika	kuantitatif	Penggunaan GCP sebagai sarana deploy container pada Kubernetes memungkinkan pengembangan aplikasi yang cepat.	Pada artikel dikatakan Docker dapat membangun sebuah aplikasi terkontainerisasi yang dapat dijalankan dimana saja. Kubernetes dapat mengatur kontainer seperti skala otomatis, monitoring, deployment, dan GCP sebagai platform untuk <i>cloud computing</i>

2.2 Profil Jabar Digital Service

2.2.1 Sejarah Perusahaan

Pemerintah Provinsi Jawa Barat (Pemprov Jabar) melalui lintas Organisasi Perangkat Daerah, lembaga dan dinas terkait, telah melaksanakan beragam proses akselerasi transformasi digital dan pengembangan ekosistem digital di Jawa Barat sejak beberapa tahun terakhir. Visi transformasi ke arah digital tersebut diinisiasi oleh Gubernur Jawa Barat, Bapak Ridwan Kamil pada awal pemerintahannya.

Keinginan kuat Gubernur untuk menciptakan Pemerintahan Berbasis Digital telah dituangkan sejak awal di dalam visi pemerintahan dalam RK3 (Ridwan Kamil, Kalla dan Kang Emil). Visi ini menegaskan kembali keinginan

kuat pemerintah provinsi untuk mendorong akselerasi digitalisasi dan reformasi birokrasi guna memberikan pelayanan publik prima kepada masyarakat Jawa Barat.

Pada Juni 2018, Bapak Ridwan Kamil secara resmi meluncurkan Jabar Digital Service (JDS) yang berfungsi sebagai *subdomain* dari jabarprov.go.id dengan tujuan memberikan kemudahan bagi masyarakat Jawa Barat dalam mengakses seluruh layanan digital yang disediakan oleh Pemprov Jabar. JDS dikelola dan dikoordinasikan oleh Dinas Komunikasi, Informatika dan Statistika (Diskominfo) Jabar.

Sejak diluncurkan, JDS terus mengembangkan berbagai inovasi layanan digital seperti Sistem Layanan Pengaduan Masyarakat (LAPOR!), Jabar Saber Hoaks, Jabar Digital Talent, Jabar Digital *Workspace*, Jabar Digital Nomads Visa, dan masih banyak lagi. JDS terus berupaya memperluas akses dan kualitas layanan digital untuk mewujudkan Provinsi Jawa Barat sebagai provinsi utama dan terbesar dalam akselerasi digitalisasi pelayanan publik di Indonesia.

2.2.2 Logo Jabar Digital Service

Logo JDS dengan huruf "J" dan "D" yang saling terkait, warna biru dan hijau, garis melengkung, dan tulisan "Service" melambangkan kolaborasi, inovasi, kepercayaan, pertumbuhan, dan komitmen JDS untuk memberikan layanan terbaik dalam membangun Jawa Barat yang maju dan digital seperti pada Gambar 2.1.



Gambar 2.1 Jabar Digital Service

2.2.3 Sapawarga



Gambar 2.2 Sapawarga

Sapawarga adalah platform berbasis mobile yang digunakan untuk layanan publik digital, informasi public terkini pemprov jabar, kegiatan rw, dan sambara atau samsat *mobile* jawa barat. Tujuan Sapawarga untuk keinginan bertransformasi menjadi provinsi digital mendorong Pemerintah Daerah Provinsi Jawa Barat berkomitmen melalui sebuah inisiatif berupa aplikasi layanan publik, Sapawarga.

2.2.4 Struktur Organisasi

Gambar 2.3 menunjukan struktur organisasi yang ada di Jabar Digital Service per tahun 2024.



Gambar 2.3 Struktur Organisasi Jabar Digital Service

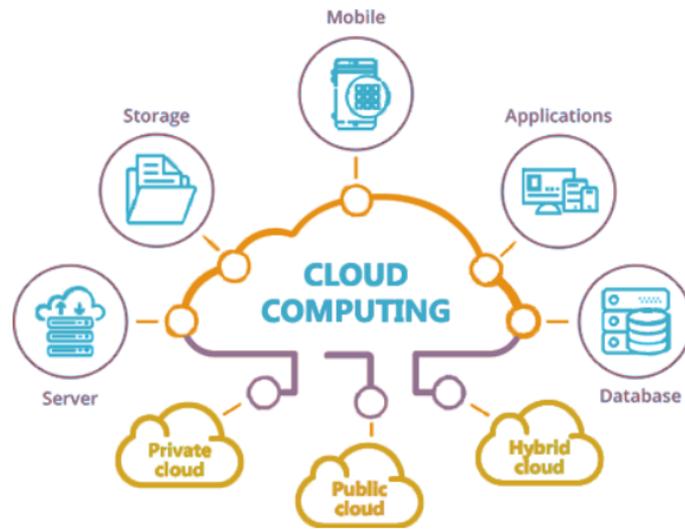
1. Unit Pelaksana Teknik Daerah (UPTD) Pelaksana Teknis Pengelola Layanan Digital, Data, dan Informasi Geospasial (PLDDIG) atau dikenal dengan Jabar Digital Service, JDS bertanggung jawab untuk mengembangkan berbagai aplikasi dan layanan digital yang digunakan oleh pemerintah dan masyarakat di Jawa Barat.

2. Sub Bagian Tata Usaha mempunyai tugas melaksanakan sebagian tugas pokok Kantor di bidang ketatausahaan yang meliputi pengelolaan kepegawaian, keuangan, perlengkapan, kerumah tanggaan dan urusan umum lainnya.
3. Jabatan Fungsional Pranata Komputer mempunyai tugas yaitu melaksanakan kegiatan teknologi informasi berbasis komputer yang meliputi tata kelola dan tata laksana teknologi informasi, infrastruktur teknologi informasi, serta sistem informasi dan multimedia
4. Jabatan Fungsional Statistisi mempunyai tugas melakukan pengumpulan, pengolahan, penyajian, penyebarluasan, dan analisis data serta pengembangan metode statistik.

2.3 Landasan Teori

2.3.1 *Cloud Computing*

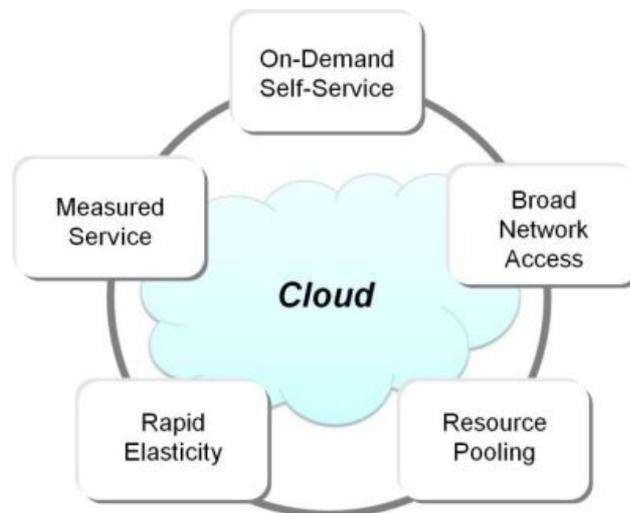
Cloud computing adalah model penggunaan sumber daya komputasi yang memungkinkan akses jaringan yang mudah dan *on-demand* ke *pool* sumber daya yang dapat disesuaikan (misalnya, jaringan, server, penyimpanan, aplikasi, dan layanan) yang dapat disediakan dan dilepaskan dengan upaya manajemen yang minimal atau interaksi dengan penyedia layanan. Cloud Computing merupakan lingkungan berbasis internet yang memungkinkan kita untuk menggunakan sumberdaya komputer dari manapun dan kapanpun melalui internet. Keuntungan dari Cloud Computing adalah Ketersediaan sumber daya, Reliabilitas serta performa yang tinggi[5]. mempromosikan ketersediaan dan terdiri dari lima karakteristik esensial, tiga model layanan, dan empat model pengembangan[6], Gambar 2.4 mempresentasikan bagaimana gambaran *cloud computing*.



Gambar 2.4 Gambaran Umum *Cloud Computing*

Sumber : nbsoft.com

2.3.1.1 Karakteristik Esensial *Cloud Computing*



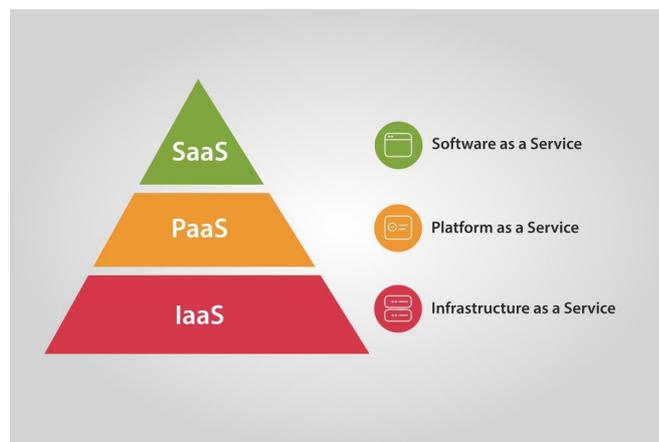
Gambar 2.5 Karakteristik Esensial *Cloud Computing*

Sumber : nesabamedia.com

Gambar 2.5 merupakan struktur karakteristik esensial *cloud computing*, berikut adalah penjelasan terkait masing-masing poin :

1. ***On-demand self-service***: Pengguna dapat meminta sumber daya secara *on-demand* dan mengaksesnya secara langsung tanpa memerlukan interaksi dengan penyedia layanan.
2. ***Broad network access***: Sumber daya dapat diakses dari mana saja dan melalui berbagai perangkat, termasuk perangkat *mobile*.
3. ***Resource pooling***: Sumber daya yang tersedia dapat dipooling dan dapat disesuaikan dengan kebutuhan pengguna.
4. ***Rapid elasticity***: Sumber daya dapat disesuaikan secara cepat dan dapat diperluas atau dikurangi sesuai dengan kebutuhan.
5. ***Measured service***: Biaya penggunaan sumber daya dapat diukur dan dikenakan berdasarkan penggunaan yang sebenarnya.

2.3.1.2 Tiga Model Layanan *Cloud Computing*



Gambar 2.6 Tiga Model Layanan *Cloud Computing*

Sumber : bridgeit.com.au

Gambar 2.6 merupakan tiga model layanan *cloud computing*, berikut adalah penjelasan terkait masing-masing poin :

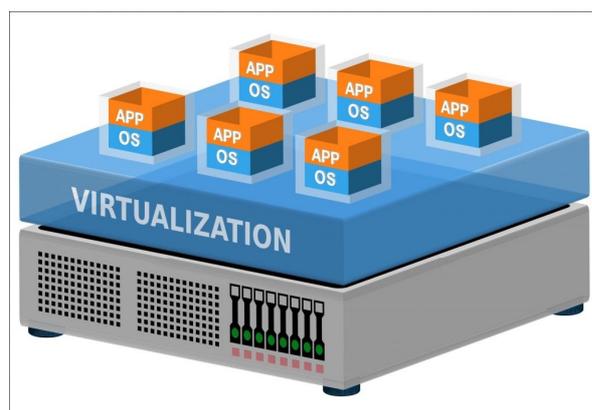
1. ***Infrastructure as a Service (IaaS)***: Penyedia layanan memberikan akses ke infrastruktur komputasi, seperti server dan penyimpanan, yang dapat digunakan oleh pengguna.

2. **Platform as a Service (PaaS):** Penyedia layanan memberikan akses ke platform komputasi yang dapat digunakan untuk mengembangkan aplikasi, seperti penyedia layanan pengembangan aplikasi.
3. **Software as a Service (SaaS):** Penyedia layanan memberikan akses ke aplikasi yang dapat digunakan secara langsung oleh pengguna, seperti email dan aplikasi produktivitas.

2.3.1.3 Empat model pengembangan *cloud computing*

1. **Private Cloud:** *Cloud computing* yang digunakan secara eksklusif oleh organisasi atau individu, biasanya di dalam jaringan organisasi.
2. **Public Cloud:** *Cloud computing* yang digunakan secara umum oleh banyak pengguna, biasanya di internet.
3. **Hybrid Cloud:** *Cloud computing* yang menggunakan kombinasi dari *private* dan *public cloud*.
4. **Community Cloud:** *Cloud computing* yang digunakan oleh beberapa organisasi yang memiliki tujuan yang sama, biasanya di dalam jaringan organisasi yang sama.

2.3.2 Virtualisasi



Gambar 2.7 Gambaran Virtualisasi

Sumber : elektro.itda.ac.id

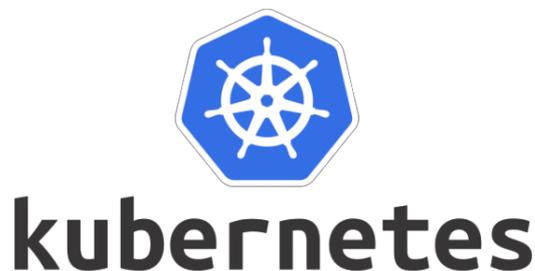
Virtualisasi adalah sebuah teknologi yang memungkinkan pengembangan aplikasi yang dapat berjalan secara independen dan seimbang

pada perangkat keras yang sama, tanpa memerlukan perangkat keras yang terpisah untuk setiap aplikasi. Docker menggunakan virtualisasi OS untuk menyimpan perangkat lunak ke dalam sebuah *container* yang terisolasi, sehingga dapat dijalankan dalam infrastruktur lokal tanpa melakukan perubahan konfigurasi pada *container*. Docker juga sangat ringan dan cepat jika dibandingkan dengan mesin virtual yang berbasis *hypervisor*, seperti pada Gambar 2.7 .

Dalam beberapa penelitian, Docker virtualisasi telah diterapkan untuk meningkatkan efisiensi dan kualitas layanan. Contohnya, dalam penelitian yang dilakukan di SMK Negeri 1 Rangkasbitung, Docker virtualisasi digunakan untuk meningkatkan efektivitas dalam penggunaan sumber daya CPU dan memori pada server. Pengujian dilakukan dengan cara melakukan variasi banyaknya jumlah *user request* yang berbeda pada masing-masing *container* menggunakan perangkat lunak apache jmeter. Hasil penelitian menunjukkan bahwa Docker virtualisasi dapat meningkatkan efisiensi penggunaan sumber daya CPU dan memori pada server.

Selain itu, Docker virtualisasi juga digunakan dalam analisis kebutuhan *resource* dan independensi antara teknologi *single server*, virtualisasi, dan *container*. Penelitian ini menunjukkan bahwa Docker virtualisasi membutuhkan memori sebesar 521 MB dan CPU 1.85%, yang lebih sedikit dibandingkan dengan *single server* dan virtualisasi. Hal ini karena Docker virtualisasi tidak memuat *kernel*, melainkan berbagi *kernel host* dengan *container* yang lain[7].

2.3.3 Kubernetes

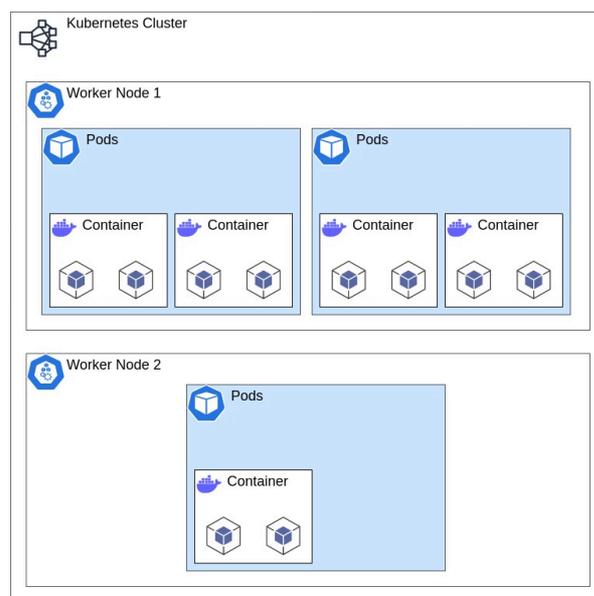


Gambar 2.8 Kubernetes

Sumber : dds.web.id

Kubernetes adalah sebuah platform manajemen kontainer yang memungkinkan pengembangan, pengelolaan, dan skala aplikasi yang berbasis kontainer secara efektif dan efisien. Kubernetes memungkinkan pengguna untuk mengelola aplikasi yang terdiri dari banyak kontainer yang berjalan secara independen dan dapat disesuaikan dengan kebutuhan aplikasi. Platform ini juga memungkinkan pengguna untuk mengelola sumber daya seperti memori, CPU, dan penyimpanan secara efektif, serta memungkinkan pengguna untuk mengelola aplikasi yang berbasis kontainer secara terintegrasi dengan sistem operasi dan infrastruktur yang lebih luas[8].

Kubernetes juga digunakan dalam beberapa penelitian untuk meningkatkan efisiensi dan kualitas layanan. Contohnya, dalam penelitian yang dilakukan di Data Center Universitas Lampung, Kubernetes digunakan untuk meningkatkan kinerja server dan menghemat biaya operasional. Pengujian dilakukan dengan cara melakukan variasi banyaknya jumlah *user request* yang berbeda pada masing-masing *container* menggunakan perangkat lunak apache jmeter. Hasil penelitian menunjukkan bahwa Kubernetes dapat meningkatkan efisiensi penggunaan sumber daya CPU dan memori pada server[9].



Gambar 2.9 Kubernetes Cluster

Sumber : latcoding.com

Selain itu, Kubernetes juga digunakan dalam analisis kebutuhan *resource* dan independensi antara teknologi *single server*, virtualisasi, dan *container*. Penelitian ini menunjukkan bahwa Kubernetes membutuhkan memori sebesar 521 MB dan CPU 1.85%, yang lebih sedikit dibandingkan dengan *single server* dan virtualisasi. Hal ini karena Kubernetes tidak memuat *kernel*, melainkan berbagi *kernel host* dengan *container* yang lain[10].

2.3.4 Nginx

Nginx adalah sebuah teknologi *frontend* web server yang digunakan untuk mengelola lalu lintas web dan mengoptimalkan kinerja server. Nginx dikembangkan untuk memenuhi kebutuhan aplikasi web yang memerlukan performa tinggi dan skalabilitas yang baik. Dalam beberapa penelitian, Nginx telah digunakan untuk meningkatkan efisiensi dan kualitas layanan. Contohnya, dalam penelitian yang dilakukan di Data Center Universitas Lampung, Nginx digunakan untuk meningkatkan kinerja server dan menghemat biaya operasional. Pengujian dilakukan dengan cara melakukan variasi banyaknya jumlah *user request* yang berbeda pada masing-masing *container* menggunakan perangkat lunak apache jmeter. Hasil penelitian menunjukkan bahwa Nginx dapat meningkatkan efisiensi penggunaan sumber daya CPU dan memori pada server[11].



Gambar 2.10 Nginx

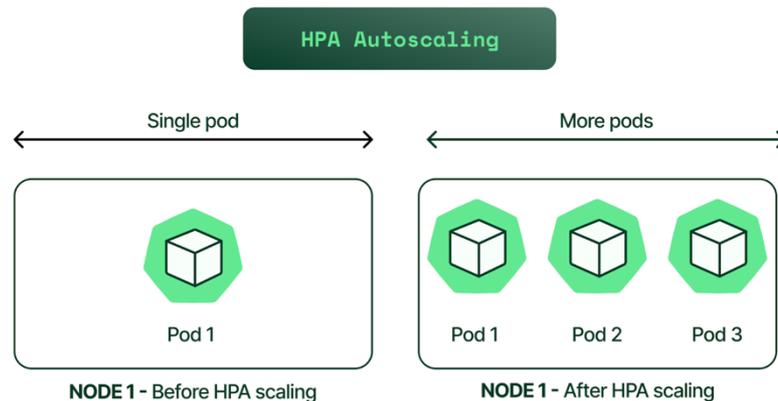
Sumber : danlogs.com

Selain itu, Nginx juga digunakan dalam analisis kebutuhan *resource* dan independensi antara teknologi *single server*, virtualisasi, dan *container*. Penelitian ini menunjukkan bahwa Nginx membutuhkan memori sebesar 521

MB dan CPU 1.85%, yang lebih sedikit dibandingkan dengan *single server* dan virtualisasi. Hal ini karena Nginx tidak memuat kernel, melainkan berbagi kernel *host* dengan *container* yang lain[12].

2.3.5 Horizontal Pods Autoscaling

Horizontal Pod Autoscaling (HPA) adalah sebuah fitur di Kubernetes yang memungkinkan pengguna untuk mengatur skala *horizontal pod* secara dinamis berdasarkan penggunaan sumber daya seperti CPU dan memori. HPA dapat membantu meningkatkan ketersediaan aplikasi dan mengurangi biaya operasional dengan mengoptimalkan penggunaan sumber daya. Dalam beberapa penelitian, HPA telah digunakan untuk meningkatkan efisiensi dan kualitas layanan, serta untuk mengatasi masalah skala aplikasi yang dinamis[13].



Gambar 2.11 Horizontal Pod Autoscaling

Sumber : [linkedin.com/saffronedge1](https://www.linkedin.com/company/saffronedge1)

Dalam beberapa penelitian, HPA telah digunakan untuk memahami bagaimana skala horizontal pod dapat membantu meningkatkan ketersediaan aplikasi dan mengurangi biaya operasional. Contohnya, dalam penelitian yang dilakukan di Data Center Universitas Lampung, HPA digunakan untuk meningkatkan kinerja server dan menghemat biaya operasional. Pengujian dilakukan dengan cara melakukan variasi banyaknya jumlah *user request* yang berbeda pada masing-masing *container* menggunakan perangkat lunak apache

jmeter. Hasil penelitian menunjukkan bahwa HPA dapat meningkatkan efisiensi penggunaan sumber daya CPU dan memori pada server, seperti pada gambar 2.11.

2.3.6 Helm

Helm Chart adalah sebuah koleksi file yang menggambarkan semua dependensi, sumber daya, dan parameter yang diperlukan untuk menginstal dan mengupgrade aplikasi di dalam sebuah *cluster* Kubernetes. Helm Chart digunakan untuk mengatur dan mengelola aplikasi yang berbasis Kubernetes dengan cara yang lebih efektif dan efisien[14].



Gambar 2.12 Helm

Sumber : iconduck.com

Helm Chart juga memungkinkan pengguna untuk mengatur dan mengelola aplikasi yang berbasis Kubernetes dengan cara yang lebih efektif dan efisien. Helm Chart juga digunakan untuk mengatasi masalah keamanan aplikasi yang berbasis Kubernetes. Contohnya, dalam penelitian yang dilakukan di Data Center Universitas Lampung, Helm Chart digunakan untuk meningkatkan keamanan aplikasi yang berbasis Kubernetes dengan cara mengidentifikasi dan mengatasi masalah keamanan yang terkait dengan aplikasi[15].

Selain itu, Helm Chart juga digunakan untuk mengoptimalkan kinerja aplikasi yang berbasis Kubernetes. Contohnya, dalam penelitian yang dilakukan di Data Center Universitas Lampung, Helm Chart digunakan untuk meningkatkan kinerja aplikasi yang berbasis Kubernetes dengan cara mengoptimalkan konfigurasi aplikasi dan mengurangi biaya operasional.

2.3.7 Prometheus



Gambar 2.13 Prometheus

Sumber : jacksonblog.medium.com

Prometheus adalah sebuah sistem monitoring yang berbasis open-source yang dikembangkan oleh SoundCloud. Dalam beberapa konteks, Prometheus digunakan untuk mengumpulkan data time-series dan memungkinkan pengguna untuk membuat visualisasi data yang dapat dikustomisasi menggunakan Grafana. Prometheus juga digunakan dalam beberapa proyek, seperti pengembangan dashboard monitoring celah keamanan pada cluster Kubernetes dengan metode container images scanning, serta dalam sistem pemantauan untuk infrastruktur jaringan yang menggunakan perangkat Ubiquity.

Dalam beberapa aplikasi, seperti analisis dan penerapan metode single exponential smoothing dan multi-agent system pada prediksi penjualan, serta penerapan FCM untuk penentuan kemampuan siswa berbasis sistem agent cerdas, Prometheus digunakan sebagai metodologi dalam pengembangan model yang berfokus pada bagaimana agen dapat membantu dalam melakukan peramalan terhadap kebutuhan stocking barang atau menentukan jenis kemampuan siswa[16].

2.3.8 Grafana

Grafana adalah sebuah alat monitoring dan visualisasi yang berbasis *open-source* yang dikembangkan untuk tujuan monitoring dan visualisasi waktu nyata. Grafana memiliki fitur-fitur yang luas dan fungsi-fungsi yang

memungkinkan organisasi untuk membuat dashboard interaktif dan dapat disesuaikan, sehingga mereka dapat memantau dan menganalisis data dari berbagai sumber dalam waktu nyata. Salah satu kekuatan utama Grafana adalah kemampuannya untuk mengintegrasikan dan visualisasi data dari sistem yang berbeda. Dengan demikian, pengguna dapat memperoleh wawasan yang holistik tentang infrastruktur, aplikasi, dan indikator bisnis mereka.



Gambar 2.14 Grafana

Sumber : manan1206.medium.com

Antarmuka pengguna yang ramah Grafana membuatnya dapat diakses oleh pengguna dengan latar belakang teknis yang berbeda. Desain yang intuitif Grafana memungkinkan pengguna untuk dengan mudah mengonfigurasi dan mempersonalisasi dashboard, memilih visualisasi yang paling relevan dan mengatur mereka untuk memenuhi kebutuhan monitoring yang spesifik. Kelebihan ini membuat Grafana solusi yang sangat fleksibel di berbagai domain, termasuk operasi IT, DevOps, bisnis intelligence, dan monitoring IoT[17].