

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Website ialah media yang terdiri dari halaman-halaman saling terhubung yang berfungsi guna menampilkan informasi dalam bentuk teks, gambar, video, atau kombinasi semuanya. Bersifat multiplatform, website dapat diakses dari bermacam perangkat yang terhubung ke internet. Meski bukan teknologi baru, banyak perusahaan masih memakainya untuk profil perusahaan, penjualan produk, atau sistem layanan bagi pelanggan [1]. Beberapa tahun terakhir, terjadi lonjakan signifikan dalam penggunaan *website* di seluruh dunia. Sesuai data yang diberi oleh similar Web, terjadi lonjakan signifikan di rentang bulan februari-maret 2024 pada situs web kategori *e-commerce* Shopee.com dengan total kunjungan sebanyak 236 juta pengunjung melonjak menjadi 275,6 juta pengunjung [2]. Dengan meningkatnya jumlah pengguna pada aplikasi web, performansi menjadi faktor kunci yang harus dipertimbangkan dengan serius.

Performansi yang buruk seperti server *down* bisa mengakibatkan bermacam dampak negatif, baik secara teknis atau bisnis maka menyebabkan penurunan penggunaan *website*. Dihimpun dari bermacam sumber media elektronik beberapa kejadian atau peristiwa server *down* yakni terjadi pada saat pembukaan rekrutment BumN dikutip dari Pikiran-rakyat.com diperkirakan ada ribuan calon pelamar yang berminat mengakses situs tersebut untuk melamar di bermacam perusahaan BUMN, namun dengan tingginya jumlah pengunjung menyebabkan situs alami keterbatasan dalam melayani pelamar yang mencoba masuk ke *platform* lamaran [3]. Selain itu juga terjadi server *down* pada *website* kpu.go.id saat berlangsungnya pemilu yang dikutip dari Pikiran-rakyat.com, Komisioner KPU Mochammad Afifuddin Afif mengatakan penyebab situs down yakni banyaknya pihak yang mengakses situs tersebut, hal itu dibenarkan melalui informasi dari jutaan orang yang mengakses situs tersebut terutama yang ingin mengetahui lokasi TPS, tetapi informasi dari jajaran sekretariat per siang tadi sebelum jam 12.00 masih bisa diakses. Pernyataan itu dibenarkan oleh pakar keamanan siber Alfons Tanujaya, ia mengatakan hal ini terjadi lantaran server KPU tidak kuat menerima *traffic* yang begitu besar [4]. Dengan adanya kejadian server *down* tersebut membuat keresahan warganet ataupun panitia KPU, maka hal yang dibutuhkan untuk mengatasi kasus tersebut dengan meningkatkan performansi dengan memakai *load balancer*.

*Load balancer* ialah teknik guna mengoptimalkan jaringan dengan membagi beban web server secara merata, memastikan traffic berjalan optimal. Metode ini mendistribusikan beban melalui dua atau lebih jalur koneksi secara seimbang dan banyak dipakai bermacam institusi serta perusahaan di seluruh dunia [5]. *Grafana* ialah aplikasi *platform opensource* monitoring yang dipakai untuk manajemen visualisasi dan analisa data dari bermacam sumber secara *real-time*. *Prometheus* ialah aplikasi yang membantu memonitor tiap aplikasi dan node yang mengeksport data metriknya ke *prometheus* untuk bisa disabilan oleh pengguna dalam bentuk HTTP [6]. Integrasi antara *Grafana* dan *Prometheus* memberi peluang pada administrator sistem untuk memantau secara *real-time* kinerja *load balancer* dan server, maka data yang dikumpulkan oleh *load balancer* bisa divisualisasikan dengan baik.

Dalam konteks pengembangan website, teknologi *load balancer* ini dipakai untuk memastikan situs web tetap responsif saat alami lonjakan *traffic* yang besar. Salah satu *framework* yang populer untuk pengembangan web ialah *Flask*, sebuah web *framework* yang ditulis dengan bahasa *Python* dan tergolong jenis *microframework* [7]. Meskipun *Flask* dikenal karena kemudahan pengoperasiannya, namun bisa dengan mudah dikembangkan menjadi aplikasi web yang kompleks dan *scalable*. Dengan memakai *Flask*, pengguna bisa membuat sebuah web yang terstruktur dan bisa mengatur *behavior* suatu web dengan lebih mudah, sambil mempertahankan kinerja dan ketersediaan yang optimal.

Sesuai penjelasan latar belakang tersebut, pada studi ini akan membahas terkait implementasi *load balancer* pada *Flask* web app untuk tingkatkan performansi web server yang bisa memberi manfaat secara signifikan dalam tingkatkan performansi, keandalan, dan skalabilitas pada website. Dengan memanfaatkan teknologi ini, pengguna bisa mempunyai kesempatan yang lebih baik dalam mengakses *website*, tanpa mengakibatkan server *down*.

## 1.2 Rumusan Masalah

Rumusan masalah di studi ini.

1. Bagaimana cara mengimplementasikan *load balancer* pada *Flask* web *application*?
2. Bagaimana melakukan monitoring dengan *Grafana* dan *Prometheus* pada *Flask* web *application*?
3. Bagaimana menganalisis perbedaan QoS pada *Flask* Web App tanpa memakai *Load balancer* dan dengan memakai *Load balancer*
4. Bagaimana menganalisis perbedaan QoS pada *Flask* Web App dengan memakai *Load balancer* saat metode *Round-robin*, *Least-connection*, dan *IP-Hash*

### 1.3 Tujuan Penelitian

Tujuan di studi ini:

1. Mengimplementasikan *load balancer* pada *Flask web application*.
2. Mampu melakukan monitoring dengan *Grafana* dan *Prometheus* pada *Flask web application*.
3. Mampu menganalisis perbedaan QoS (*Throughput, Packet Los, Jitter, Delay*) pada *Flask Web App* tanpa memakai *Load balancer* dan dengan memakai *Load balancer*
4. Mampu menganalisis perbedaan QoS (*Throughput, Packet Los, Jitter, Delay*) pada *Flask Web App* dengan memakai *Load balancer* saat metode *Round-robin, Least-connection, dan IP-Hash*

### 1.4 Batasan Masalah

Batasan masalah di studi ini.

1. *Load balancer* yang dipakai ialah *Nginx load balancer* dengan metode *Round-robin, least-connection, dan IP-Hash*.
2. Web yang dipakai, dirancang dengan *framework python Flask* dan di-deploy pada *Ubuntu server*.
3. Cakupan *Quality of Service (QoS)* yang dianalisa ialah *throughput, delay, Jitter, dan packet loss*.

### 1.5 Manfaat Penelitian

Studi ini diharapkan bisa memberi manfaat yang signifikan dalam meningkatkan performansi web server memakai *load balancer*, hal ini bisa didistribusikan secara merata diantara beberapa server backend, dengan mengurangi beban pada server tunggal dan mempercepat waktu tanggapan aplikasi. Dengan otomatis mengarahkan lalu lintas ke server yang masih berfungsi, *load balancer* juga meningkatkan tingkat ketersediaan aplikasi, mengurangi risiko *downtime* dan memastikan pengalaman pengguna yang lebih baik secara keseluruhan.

### 1.6 Sistematika Penulisan

Dalam penulisan Skripsi tersusun atas lima bab, yakni.

## BAB I PENDAHULUAN

Memuat latar belakang, rumusan masalah, batasan masalah tujuan dan manfaat, serta sistematika penulisan.

## BAB II DASAR TEORI

Memuat teori pendukung pengerjaan Skripsi, seperti penjelasan terkait *Load balancer, Grafana dan Prometheus, Quality of Service (throughput, delay, Jitter, dan packet loss)*.

### **BAB III METODOLOGI PENELITIAN**

Memuat Metodologi Penelitian Skripsi, alur pengerjaan Skripsi, identifikasi data, serta pengujian dan pengambilan data.

### **BAB IV PENGUJIAN DAN ANALISIS**

Memuat pengujian, analisis, dan pembahasan terkait fungsi *load balancer*, monitoring web server, dan *quality of service*

### **BAB V KESIMPULAN DAN SARAN**

Memuat kesimpulan dan saran dari pengerjaan Skripsi untuk pembaca yang akan mengambil penelitian dengan topik yang sama.