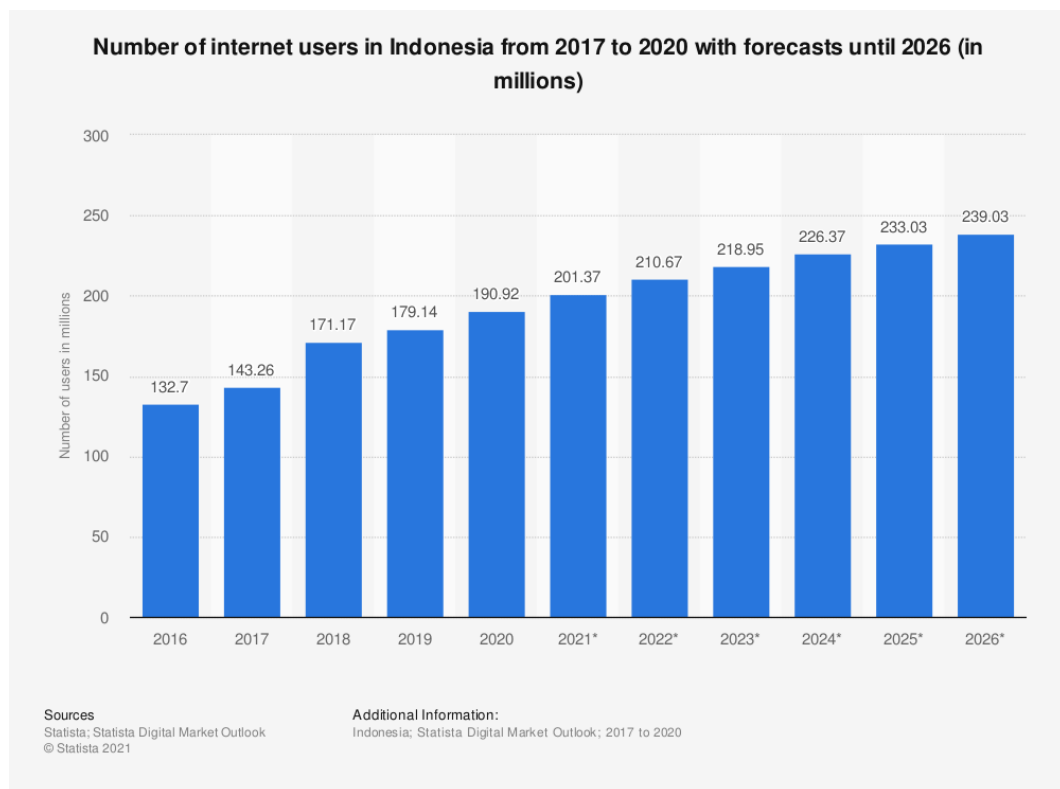


BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pemanfaatan teknologi informasi sudah sangat luas hingga ke dalam kehidupan sehari-hari manusia. Setiap saat manusia tidak bisa lepas dengan perangkat elektronik yang disebut *gadget* seperti ponsel pintar, laptop, tablet dan sebagainya yang setiap saat terhubung ke internet untuk bisa memanfaatkan fungsinya dengan maksimal. Menurut data statistik pada gambar 1.1 dari [statista.com](https://www.statista.com) peningkatan pengguna internet di Indonesia diperkirakan hingga tahun 2026 akan selalu naik sehingga akan meningkatkan *trafik* internet [1].



Gambar 1.1 Statistik pengguna internet di Indonesia [1]

Dalam dunia pendidikan internet sudah berkembang secara luas dan maju ditandai dengan metode pembelajaran yang di dukung oleh teknologi informasi. Proses belajar mengajar secara daring dapat membuat proses pembelajaran dan pemberian materi menjadi lebih interaktif dengan adanya platform *E-Learning*

untuk menjembatani pengajar dengan siswa [2]. Ketika pengguna *E-Learning* meningkat maka beban yang diberikan kepada server *E-Learning* juga akan meningkat dan ketika server tidak mampu menangani akses pengguna maka akan menjadi masalah. Oleh karena itu, maka dibutuhkan sebuah sistem server yang dapat menangani banyak akses sekaligus dan dapat di tingkatkan skalabilitasnya sesuai dengan kebutuhan. Sistem yang dapat memenuhi kebutuhan tersebut adalah server *load balancing* [3].

Server merupakan sistem komputer yang bertugas menyimpan data berupa informasi maupun berkas-berkas dokumen. Server memberikan layanan mengirim dan menerima data kepada klien yang membutuhkannya. Server memiliki fungsinya masing-masing, sebagai contoh *web server* memiliki fungsi untuk menyediakan layanan berupa data halaman web, suara, gambar yang diminta oleh *user/klien* melalui *web browser* seperti Chrome dan Firefox [4]. Salah satu perangkat lunak yang banyak digunakan untuk membangun atau menyediakan *website* adalah Apache atau lebih tepatnya Apache HTTP Server [5]. Server tersebut pastinya memiliki banyak pengguna atau pengakses di seluruh dunia, sebuah server dalam suatu perusahaan tidak mungkin hanya memiliki satu *server* fisik, tapi banyak *server* di satu tempat/lokasi yang berguna untuk menangani pengguna internet yang sangat banyak. Dalam mengurus distribusi permintaan klien ke *server*, digunakanlah sebuah sistem yang disebut *load balancing* [6].

Load balancing merupakan sebuah teknik ketika *trafik* dari permintaan klien yang datang dari dua atau lebih jalur koneksi ke *server* di distribusikan secara merata ke *server* yang tersedia sehingga *server* dapat bekerja melayani klien secara optimal tanpa terjadi *overload* atau kelebihan beban [7]. Sistem *load balancing* dibuat menggunakan algoritma penjadwalan yang di rancang untuk menangani *trafik* yang datang sesuai dengan kondisi yang ditentukan dalam kode program yang terpasang [8]. Sebuah server *load balancer* menggunakan sebuah perangkat lunak yang mengatur beban ke *web server* sesuai konfigurasi atau algoritma yang dipasang, salah satunya adalah Haproxy yang akan digunakan di penelitian ini. Beberapa algoritma yang umum digunakan pada *load balancing server* adalah *least connection* dan *round robin* [7]. Algoritma *round robin* memiliki turunan yang

memiliki performa yang lebih unggul, bernama *weighted round robin (WRR)* sementara *least connection* memiliki turunan yaitu *weighted least connection (WLC)*. Algoritma *weighted round robin* dan *weighted least connection* memiliki keunggulan dibandingkan pendahulunya [10].

Pengukuran performa sistem *load balancing server* bisa dilakukan dengan menghitung *throughput, response time, packet lost dan bandwidth*. Penghitungan dilakukan dengan melakukan pengujian akses dengan banyak koneksi dari klien ke *server* setelah itu dilakukan pengukuran [11]. Dengan adanya pengujian ini, diharapkan kedepannya bisa memilih algoritma yang terbaik dari penelitian ini sesuai dengan studi kasusnya.

1.2. Perumusan Masalah

Algoritma *load balancing* yang handal diperlukan untuk membuat sebuah *load balancing server* yang dapat menangani *request* dari klien dengan baik, oleh karena itu diperlukan perbandingan performa algoritma *load balancing weighted round robin* dan *weighted least connection* untuk mencari mana yang lebih baik.

1.3. Pertanyaan Penelitian

1.3.1. Bagaimana cara mengukur performa dari *load balancing web server*?

1.3.2. Bagaimana cara membandingkan performa algoritma *load balancing WRR dan WLC*?

1.4. Batasan Masalah

Untuk menghindari pembahasan yang terlalu luas, maka berikut ini batasan masalah penelitian ini:

1.4.1. Parameter yang akan diukur dibatasi pada *throughput, response time, packet loss dan bandwidth*,

1.4.2. Server menggunakan sistem operasi CentOS dengan web server Apache,

1.4.3. Tidak menangani keamanan dalam membangun server,

1.4.4. Pengukuran menggunakan *software* Httperf dan Iperf,

1.4.5. Algoritma yang digunakan hanya dua, yaitu *weighted round robin* dengan *weighted least connection*.

1.4.6. Menggunakan satu server basis data.

1.5. Tujuan Penelitian

1.5.1. Mengukur performa *load balancing web server* dalam menangani beban *trafik* dari klien.

1.5.2. Membandingkan performa algoritma *load balancing server* antara *WRR* dengan *WLC*.

1.6. Manfaat Penelitian

Penelitian ini diharapkan dapat membantu dalam pembuatan server yang memiliki kemampuan menangani banyak pengakses tanpa harus mengalami *overload* dengan pemilihan algoritma *load balancing* yang sesuai kebutuhan dari kedua algoritma *load balancing weighted round robin* dan *weighted least connection*.