

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*Server Maintenance* dilakukan untuk menjaga agar kinerja server menjadi lebih optimal dan berjalan maksimal maupun untuk mencegah kerusakan *Hardware* maupun *Software*. Sebagai *Network Administrator* (NA) maupun *System Administrator* harus melakukan *maintenance* untuk menjaga informasi terbaru dan menghapus informasi yang tidak dibutuhkan atau usang maupun data yang berada pada sistem [?]. Salah satu *maintenance server* yang dilakukan adalah masuk satu persatu ke server dan memeriksa *log* setiap server. Cara ini dianggap kurang efisien sehingga perlu dibuat sebuah sistem yang dapat digunakan untuk memonitor *log* setiap *server* secara terpusat. Sistem yang dibuat harus dapat menyimpan *log* yang dikirimkan *server* maupun peralatan jaringan ke *database* ditampilkan di *website* dan dapat melakukan pencarian *log* dengan cepat sehingga proses *troubleshoot* bisa dilakukan secepatnya [?]. Sistem harus mempunyai *response time* yang kecil. Paling tidak dibutuhkan waktu kurang dari 1 menit untuk menghasilkan data pencarian. Sedangkan untuk pencarian 1 juta data pada sistem yang sekarang dibutuhkan waktu kurang lebih 1,2 menit dengan *query* pencarian salah satu *hostname*.

*Log events* menghasilkan data setiap hari yang sangat bernilai [?]. Universitas Telkom (Tel-U) merupakan perguruan tinggi swasta yang sekarang memiliki puluhan ribu mahasiswa dengan 7 fakultas. Setiap data mahasiswa, dosen, maupun pegawai dari Tel-U disimpan didalam data center yang di kelola oleh Direktorat Sistem Informasi (Sisfo) Tel-U. Tentu saja dengan banyaknya data maupun layanan yang ada seperti aplikasi mahasiswa, registrasi, maupun *mail server* dibutuhkan penyimpanan yang besar untuk menampung semua data yang masuk. Terhitung sampai dengan sekarang ini terdapat 56 server dan beberapa perangkat jaringan yang digunakan oleh Sisfo untuk mengelola seluruh data Tel-U. *Server* menghasilkan *log* dengan jumlah yang sangat besar setiap harinya [?].

Solusi yang akan dicoba diterapkan untuk mengatasi pencarian data adalah menggunakan arsitektur perancangan sistem dengan menggunakan *log server* dan *solr server*,

sedangkan untuk proses pencarian mengkombinasikan antara Solr dan MongoDB. MongoDB merupakan *database* NoSQL sedangkan Solr merupakan *search platform*. Di-harapkan dengan menggunakan MongoDB dan Solr proses *searching* data akan menjadi lebih cepat, tepat, dan efisien. Hal tersebut akan diukur melalui *response time* dan *cpu resource* yang dihasilkan setelah menggunakan Apache Solr pada server yang sama.

## 1.2 Perumusan Masalah

Rumusan masalah yang menjadi acuan Tugas Akhir ini adalah :

1. Bagaimana membangun sistem *monitoring log* untuk memantau *log* setiap *server* maupun perangkat jaringan di Universitas Telkom sehingga proses pencarian *log* tidak dilakukan secara manual satu persatu masuk ke dalam server?
2. Bagaimana mendesain sistem dengan menggunakan MongoDB dan Apache Solr sehingga proses pencarian *log* pada sistem lebih ringan dan cepat ?
3. Bagaimana kinerja pencarian dengan menggunakan Solr di *server* yang sama diukur dari *response time* dan *cpu resource*?

## 1.3 Tujuan Penelitian

Tujuan dari penulisan Tugas Akhir ini adalah :

1. Mengimplementasikan MongoDB dan Apache Solr ke dalam sistem *log server* untuk melakukan pencarian *log*.
2. Menjelaskan desain sistem *log server* dengan menggunakan MongoDB dan Apache Solr.
3. Menjelaskan mengenai *response time* pencarian sistem yang dibangun dengan menggunakan MongoDB dan Solr.

## 1.4 Batasan Masalah

Adapun batasan masalah dari Tugas Akhir ini adalah :

1. Aplikasi yang dikembangkan hanya pada bagian pencarian data berdasarkan tanggal, nama *host*, *IP address*, dan *Messages*.
2. Karena ada batasan dengan penggunaan perangkat di Sisfo maka data yang digunakan adalah data dari 7 perangkat jaringan dan server yang ada di lingkungan Tel-U, yaitu server juniper, student, dan beberapa server penyimpanan.
3. Dalam pengujian, sistem dikondisikan hanya menjalankan *service* Solr, MongoDB, dan Rsyslog.
4. Sistem dibangun pada *server* yang sama.

## 1.5 Metodologi Penyelesaian Masalah

Metodologi yang digunakan pada penulisan tugas akhir ini adalah:

1. Studi Literatur Mempelajari teori – teori yang berhubungan dengan tugas akhir. Penulis mencari pengetahuan mengenai Solr sebagai aplikasi *search platform*, MongoDB untuk penyimpanan data, format data JSON, Rsyslog sebagai aplikasi *log server*, dan skema yang akan digunakan dalam perancangan sistem.
2. Analisis dan Perancangan Sistem Pada tahap ini dilakukan analisa terhadap sistem informasi log server yang sudah ada kemudian dari analisa tersebut dilakukan perancangan Solr yang ditempatkan diantara *web server* dan *database*. Kemudian dilakukan analisis terhadap data yang akan diujikam dan optimasi yang akan dihasilkan.
3. *Testing* dan analisis hasil Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun. Pengetesan yang akan dilakukan adalah dengan melakukan pencarian data *log* berdasarkan teks. Kemudian pencarian selanjutnya adalah dengan satu kata kunci yang melibatkan keseluruhan data yang tersimpan di *database*.

4. Pembuatan laporan Pada tahap ini dilakukan pembuatan laporan dari hasil analisa akhir yang dilakukan. Pada laporan akan dijabarkan semua dengan detail bagaimana penelitian dilakukan dan memberikan kesimpulan dari hasil analisa akhir.

## 1.6 Hipotesa

Data *log* yang digunakan berjumlah puluhan juta. Pencarian dari *http response* langsung ke MongoDB membutuhkan waktu yang lama. Solr merupakan *search platform* yang dapat menjembatani antara *web server* dengan *database*. Solr memberikan solusi pencarian berbasis web. Diperkirakan jika pencarian dilakukan dengan menggunakan Solr akan menghasilkan data pencarian yang cepat. Hal ini dikarenakan Solr mempunyai dua dasar pencarian, yaitu *Query Boolean* dan *Phrase Boolean*. Salah satu fitur *query* yang dapat digunakan adalah *facet*. Solr juga memanfaatkan *cache* sistem operasi. Hal ini bertujuan untuk menghindari pembacaan dari *disk* dan juga penggunaan proses [?]. Jadi index pada Solr tidak disimpan pada *disk* tetapi disimpan di RAM sehingga Solr tidak perlu membaca disk untuk melakukan pencarian. Sehingga besar alokasi RAM yang digunakan untuk Solr juga berpengaruh terhadap kinerja dari Solr. Selain itu semakin banyak data mengakibatkan ruang lingkup pencarian semakin membesar sehingga dimungkinkan dapat memperbesar *response time* dari pencarian Solr. Dimungkinkan juga akan ada perbedaan *response time* jika memvariasikan *query* karena ada beberapa *query* yang diperkirakan mengakibatkan *response time* menjadi lebih lama.

## 1.7 Rencana Penelitian

Rencana penelitian ini direncanakan akan seperti berikut :

