



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Instant messaging belakangan bukan lagi merupakan teknologi yang asing di kalangan masyarakat. *Instant messaging*, perangkat lunak aplikasi yang merupakan pengembangan dari teknologi komputer dan internet, diciptakan sebagai alat yang membantu komunikasi manusia. *Instant messaging* dapat didefinisikan sebagai sebuah bentuk komunikasi yang terdiri dari pertukaran pesan-pesan teks secara sinkronis antara pengguna melalui perangkat antarmuka berbasis web (Witts, 2009). Popularitas *instant messaging* terus meningkat karena merupakan bentuk komunikasi yang cepat dan praktis.

Instant messaging awalnya populer sebagai media komunikasi yang bersifat *casual* dan digunakan untuk kebutuhan komunikasi pribadi. Namun sekarang, pemanfaatan *instant messaging* ini meluas sampai digunakan oleh perusahaan-perusahaan (*enterprise*) sebagai penunjang kegiatan operasional sehari-hari perusahaan. Keberadaan surat elektronik intra perusahaan, mulai banyak digantikan dengan penggunaan *enterprise instant messaging*.

Kebutuhan perusahaan akan komunikasi dengan *instant messaging* juga dapat dilihat seiring perkembangan teknologi *cloud computing*. Bagi banyak perusahaan, teknologi *cloud computing* dimanfaatkan sebagai media kolaborasi. Dikembangkannya perangkat lunak berbasis *cloud computing* yang dibuat untuk proses kolaborasi memungkinkan karyawan atau partisipan dalam suatu proyek untuk dapat bekerja di balik meja kerja masing-masing tanpa harus bertemu secara

langsung. Pada aplikasi *collaborative cloud computing* seperti Google Docs, telah disisipkan fitur berupa aplikasi *instant messaging* untuk mempermudah kolaborasi, dan pada kenyataannya, fitur *instant messaging* ini sangat mendukung produktivitas pengguna layanan dalam berkolaborasi.

Kolaborasi dengan teknologi *cloud computing* tentu dilakukan oleh lebih dari satu partisipan. Dengan jumlah partisipan yang tidak dibatasi secara eksplisit oleh sistem, terlalu banyaknya pengguna dapat membebani kinerja *web server*. Oleh karena itu, diperlukan teknologi yang dapat mengakomodasi banyak pengguna dalam mengakses fitur *instant messaging* tanpa menurunkan kinerja aplikasi *cloud* tersebut.

Event-driven web server adalah sebuah solusi bagi *web server* yang memerlukan tingkat *concurrency* tinggi. *Event-driven* berarti proses yang merespon terhadap terjadinya sebuah *event*. *Event* didefinisikan sebagai suatu kejadian atau sesuatu yang terjadi dalam sistem yang biasanya memiliki atribut seperti waktu kejadian, lokasi komponen, dan data yang menggambarkan kejadian tersebut (Carbou, 2011).

Node.js adalah satu contoh *platform* yang mengimplementasikan *event-driven web server*. Node.js dikembangkan oleh Ryan Dahl dengan tujuan awal menciptakan *website* dengan kemampuan *push*. Node.js dibangun di atas bahasa javascript yang bersifat *event-driven* sehingga memungkinkan penulisan kode untuk operasi yang bersifat *non-blocking*. Dengan kemampuan tersebut, *web server* yang dibangun dengan Node.js mampu menangani *connection request* dalam jumlah tinggi lebih efisien daripada *web server* yang berbasis *thread*.

Selain dari masalah kinerja *server*, kinerja dari aplikasi *groupware* yang bersifat *real-time* banyak mendapat kritik. Kebanyakan dari kritik tersebut adalah kritik yang berhubungan dengan masalah jaringan, antara lain *latency*, *jitter*, *loss*, dan *bandwidth*. Masalah-masalah ini adalah masalah yang umum ditemukan pada arsitektur jaringan seperti *Wide Area Network* dan jaringan internet. Dari 4 masalah tersebut, keterbatasan *bandwidth* adalah faktor yang cukup berpengaruh dalam kinerja jaringan, karena ukuran dan efisiensi dari pesan yang dikirimkan pada *groupware* memegang peranan penting dalam meningkatkan kinerja aplikasi *groupware* tersebut.

Kompresi data adalah salah satu cara yang dapat mengurangi ukuran dan meningkatkan efisiensi dari pesan pada aplikasi *groupware*. Dengan kompresi data, ukuran pesan dapat dikurangi dan penggunaan *bandwidth* pada pengiriman data dalam jaringan dapat menjadi lebih efisien.

LZ77 adalah salah satu teknik kompresi yang merupakan cikal bakal dari banyak varian algoritma-algoritma kompresi lainnya. Algoritma LZ77 dibuat oleh Abraham Lempel dan Jacob Ziv pada tahun 1977. Teknik kompresi ini merupakan teknik kompresi yang bersifat *lossless* sehingga tepat digunakan untuk pengiriman pesan, sehingga tidak ada sedikitpun informasi yang hilang atau berubah selama proses kompresi, pengiriman, dan dekompresi data. Teknik kompresi LZ77 ini dapat dikatakan optimal jika input *string* merupakan runtutan karakter yang acak (*random*), stasioner, *ergodic*, dan mengandung nilai-nilai dari alfabet.

Berdasarkan latar belakang tersebut, penulis bermaksud mengimplementasikan algoritma LZ77 untuk kompresi data *instant messaging*

yang dibangun pada *event-driven web server*.

1.2 Perumusan Masalah

Masalah yang dirumuskan dalam penelitian ini adalah: “Bagaimana mengimplementasikan teknik kompresi LZ77 pada aplikasi *instant messaging* yang dibangun pada *event-driven web server*?”

1.3 Batasan Masalah

Penelitian akan dilakukan dengan berfokus pada implementasi teknik kompresi LZ77 pada aplikasi *instant messaging* pada *event-driven web server* yang dibangun di atas *platform* Node.js. Pembatasan masalah lainnya dalam penelitian ini meliputi hal-hal sebagai berikut.

1. Penelitian dilakukan dengan menggunakan teknologi websocket pada event-driven web server dengan platform Node.js.
2. Penelitian hanya akan dilakukan pada data berupa *string*.
3. Aplikasi yang dibangun merupakan aplikasi *instant messaging* berbasis *web* dengan fitur *group chat* dan indikator *presence* (kehadiran).
4. Aplikasi yang dibangun tidak meliputi pengembangan *groupware* atau aplikasi kolaborasi.
5. Penelitian tidak mencakup aspek keamanan dari pengiriman pesan.

1.4 Tujuan Penelitian

Penelitian bertujuan untuk mengimplementasikan algoritma LZ77 pada aplikasi *instant messaging* berbasis *web* yang dibangun pada *event-driven web server*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah meningkatkan efisiensi penggunaan *bandwidth* jaringan dalam proses pengiriman informasi melalui aplikasi *instant messaging*.

1.6 Sistematika Penulisan

Laporan ini tersusun menjadi beberapa bab dengan penjelasan masing – masing bab adalah sebagai berikut.

Bab I Pendahuluan

Bab ini berisi latar belakang permasalahan, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan laporan.

Bab II Landasan Teori

Bab ini berisi teori – teori terkait dengan pelaksanaan penelitian ini. Teori – teori yang digunakan antara lain adalah teori mengenai *instant messaging*, konsep kolaborasi, kompresi data, arsitektur *event-driven*

Bab III Analisis dan Perancangan Aplikasi

Bab ini berisi analisis penulis mengenai penerapan algoritma kompresi

LZ77 pada pembangunan aplikasi *instant messaging*, perancangan dan implementasi antarmuka aplikasi, implementasi kode-kode pembangun aplikasi yang dibangun pada *event-driven web server*, *flowchart* yang menunjukkan alur kerja program.

Bab IV Uji Coba dan Pembahasan

Bab ini berisi hasil uji coba terhadap aplikasi beserta analisis pembahasan oleh penulis dari hasil yang diperoleh tersebut.

Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan oleh penulis mengenai aplikasi yang telah dibangun secara keseluruhan, serta beberapa saran yang dapat diaplikasikan untuk pengembangan aplikasi selanjutnya.

UMMN