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.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil implementasi dan uji coba yang telah dilakukan, simpulan penelitian implementasi algoritma Deflate pada aplikasi pengiriman dan penerimaan *file* dengan menggunakan teknologi Android Beam adalah implementasi dapat dilakukan pada aplikasi pengiriman dan penerimaan *file* dengan menggunakan Android Beam dan telah mencapai tujuan penelitian. Temuan dan analisis yang didapat pada penelitian adalah sebagai berikut.

1. Aplikasi berjalan dengan baik ketika melakukan kompresi, pengiriman, dan dekompresi. Selain itu hampir tidak terjadi *error* atau *bug* pada aplikasi ketika sedang digunakan. Namun pengaruh komposisi *file* yang akan diproses dan perangkat yang digunakan berpengaruh dalam kinerja dan efektivitas aplikasi.
2. Berdasarkan hasil implementasi dan uji coba kompresi dapat disimpulkan *file* yang tidak mengalami format kompresi memiliki ukuran kompresi yang lebih signifikan dari pada *file* yang telah terformat kompresi lainnya. *File* yang bertipe *.txt, *.doc, dan *.tga dapat dikompres dengan baik. Sebaliknya *file* yang bertipe *.jpg/jpeg dan *.mp3 tetap dapat terkompres namun ukuran hanya berkurang sedikit. Waktu kompresi juga tidak terlalu dirasakan pengguna karena waktu kompresi yang singkat (berlangsung dalam milidetik) dengan tingkat kegagalan kompresi 0%.

Implementasi Algoritma ..., Yuddistira Kiki, FTI UMN, 2014

3. Berdasakan hasil implementasi dan uji coba pengiriman menggunakan Android Beam dapat terlihat dengan jelas waktu pengiriman untuk *file* yang sudah terkompresi lebih singkat dari pada waktu pengiriman untuk *file* tanpa kompresi. Selain itu pengujian juga membuktikan waktu pengiriman sangat dipengaruhi oleh komposisi *file*, proses koneksi antar perangkat yang dapat terjadi dengan cepat atau lambat, serta performa perangkat saat melakukan proses pengiriman.
4. Proses dekompresi terlihat sesuai dengan tujuan dan spesifikasi berdasarkan hasil uji coba. Proses dekompresi diaplikasi berhasil mengembalikan *file* yang terkompresi menjadi *file* original atau *file* asli dengan tingkat kegagalan 0%. Waktu dekompresi dipengaruhi dengan ukuran *file* dan perangkat yang digunakan. Namun sama hal seperti kompresi, waktu dekompresi tidak terlalu dirasakat karena berlangsung dengan sangat singkat.
5. Dari semua uji coba, analisis, dan evaluasi yang didapat dapat disimpulkan implementasi algoritma Deflate pada aplikasi pengiriman dan penerimaan *file* dengan menggunakan Android Beam berhasil memberikan perkembangan dalam proses pengiriman menggunakan Android Beam. Dengan terkompresinya *file* yang akan dikirim, memberikan waktu pengiriman yang lebih singkat.

5.2 Saran

Beberapa saran yang dapat dijadikan pedoman untuk penelitian selanjutnya yaitu sebagai berikut.

1. Dengan melihat implementasi algoritma Deflate berhasil diterapkan pada Android Beam maka ada kemungkinan penggunaan algoritma – algoritma kompresi baru yang lebih baik dan lebih cocok diimplementasikan pada Android Beam. Hal tersebut yang dapat dijadikan pokok bahasan pada penelitian selanjutnya.
2. Terbatasnya fungsi aplikasi yang melakukan kompresi, pengiriman, dan dekompresi membuat aplikasi memiliki fungsi penggunaan yang tidak banyak. Kedepannya dapat dikembangkan kembali aplikasi yang memiliki fungsi lebih banyak dan baik seperti dapat melakukan kompresi, pengiriman, dan dekompresi untuk *multi file* atau banyak *file* yang diproses.
3. Pengaruh perangkat keras dalam proses yang terjadi di aplikasi sangatlah berpengaruh. Dengan semakin besarnya ukuran *file* semakin besar *resource* perangkat yang akan digunakan. Dengan terbatasnya keterbatasannya perangkat maka diperlukan perangkat yang lebih baik. Solusi lainnya adalah mengembangkan aplikasi yang lebih efisien dan efektif dalam melakukan prosesnya.