

Algoritma Kompresi LZ77 dan Huffman untuk Media Sosial Berbasis Android Studi Di Universitas Budi Luhur

Fransisco¹⁾, Rizky Tahara Shita²⁾

¹Program Studi Teknologi Informasi, Fakultas Teknologi Informasi, Universitas Budi Luhur

^{1,2}Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

E-mail : siscofran999@gmail.com¹⁾, rizky.tahara@gmail.com²⁾

Abstrak

Di Universitas Budi Luhur mahasiswa atau warga kampus kurang mengetahui akan adanya acara yang akan diselenggarakan di kampus, Hal ini disebabkan oleh media penyampaian informasi yang ada seperti baliho atau spanduk kurang efektif dalam mempromosikan event yang akan berlangsung. Di website resmi Universitas Budi Luhur pun, hanya event-event besar dan yang memiliki hubungan langsung dengan kampus yang akan ditampilkan. Di media sosial, seperti Instagram ataupun Path informasi yang tersedia didalamnya masih bersifat umum, belum khusus Universitas Budi Luhur. Terkadang, dosen juga ikut mempromosikan event tersebut dengan mengajak mahasiswa mengikuti event dengan harapan mahasiswa akan tertarik dan acara yang diselenggarakan. Tujuan dari penelitian ini agar warga kampus mengetahui kegiatan / event yang berada di kampus dengan berbagi gambar seperti PNG, JPEG, BMP dan dapat berbagi informasi berupa file PDF (Portable Document Format) kepada teman maupun rekan kerja di kampus. Dengan algoritma kompresi LZ77 dan Huffman sebagai kompresi file gambar dan PDF, aplikasi dibangun berbasis android dengan menggunakan React-Native dengan bahasa pemrograman yaitu javascript dan xampp sebagai web server. Proses pengiriman postingan akan melalui proses kompresi dengan menggunakan algoritma LZ77 dan Huffman yang kemudian di simpan di server, sedangkan proses menampilkan postingan akan melalui proses dekompresi dengan menggunakan algoritma LZ77 dan Huffman sebelum postingan dimunculkan. Berdasarkan hasil pengujian, ternyata algoritma LZ77 jika berdiri sendiri memiliki proses kompresi dan dekompresi yang lebih cepat dari algoritma Huffman, namun dari segi ukuran file yang di compress algoritma Huffman memiliki rasio hasil kompresi yang lebih kecil dari pada algoritma LZ77. Jika algoritma LZ77 dan Huffman digabung maka waktu yang di butuhkan untuk melakukan proses kompresi dan dekompresi lebih cepat dari pada algoritma Huffman jika berdiri sendiri, namun lebih lambat dari algoritma LZ77 jika berdiri sendiri, juga memiliki rasio yang lebih kecil dari pada algoritma Huffman dan LZ77 jika berdiri sendiri, dengan rata-rata rasio sebesar 87,9 % untuk file PDF, 100,2 % untuk file PNG, 100,1% untuk file JPEG, dan 48% untuk file BMP.

Kata kunci : Media Sosial, Kompresi , LZ77 , Huffman, React-Native.

1. PENDAHULUAN

Media Sosial awalnya bermula dari kebutuhan pengguna internet yang ingin berkomunikasi dengan orang lain yang berbeda tempat, kemudian berkembang menjadi tempat berbagi informasi kepada orang lain yang ingin mengetahui informasi tersebut. Melalui aplikasi media sosial, manusia bisa berbagi informasi kepada teman, sahabat, maupun keluarga. Begitu juga di Universitas Budi Luhur, kebutuhan mahasiswa atau warga kampus kurang mengetahui akan adanya acara yang akan diselenggarakan di kampus, Hal ini disebabkan oleh media penyampaian informasi yang ada seperti baliho atau spanduk kurang efektif dalam mempromosikan event yang sedang atau akan berlangsung. Di website resmi Universitas Budi

Luhur pun, tidak semua event ada di tampilan, hanya event-event besar dan yang memiliki hubungan langsung dengan kampus yang akan ditampilkan. Di media sosial, seperti Instagram ataupun Path informasi yang tersedia didalamnya masih bersifat umum, belum khusus Universitas Budi Luhur. Terkadang, dosen juga ikut mempromosikan event tersebut dengan mengajak mahasiswa mengikuti event dengan harapan mahasiswa akan tertarik dan acara yang diselenggarakan akan ramai. Maka dari itu, aplikasi media sosial ini dibuat bertujuan agar warga kampus mengetahui kegiatan atau event yang berada di kampus dengan berbagi gambar dan dapat berbagi informasi berupa file PDF (Portable

Document Format) kepada teman maupun rekan kerja di kampus, juga sebagai sarana mencari teman baru dengan tetap memaksimalkan ukuran dan kualitas gambar dan *file* PDF yang ada agar tidak membebani media penyimpanan.

2. METODE PENELITIAN

Metode Penelitian yang digunakan adalah metode *Waterfall*. Penelitian harus berasal dari sumber data yang lengkap dan terpercaya, dikarenakan dibutuhkan untuk dokumentasi penulisan karya ilmiah dan menentukan analisa penelitian untuk memecahkan masalah yang ingin diselesaikan. Pengumpulan data ini menggunakan proses pencarian data dengan cara mencari *literature review*, jurnal ilmiah, buku-buku dan bacaan-bacaan yang berhubungan dengan judul penelitian ini.

a. Analisa Kebutuhan.

Dalam tahap ini diperlukan adanya analisa terhadap kebutuhan sistem, cara pencarian jurnal ilmiah, buku-buku dan sumber referensi lainnya yang berkaitan dengan penulisan penelitian ini untuk mengumpulkan data dan menyelesaikan masalah yang terkait.

b. Desain Sistem.

Tahapan desain bertujuan untuk menentukan spesifikasi detil dari komponen-komponen program seperti manusia, perangkat keras, perangkat lunak, jaringan, data dan produk-produk informasi lainnya yang sesuai dengan hasil tahapan analisis yang kemudian digambarkan ke dalam bentuk grafik atau diagram yang mudah dimengerti, sehingga dapat dijelaskan dengan detail setiap fungsi dan kebutuhan yang diinginkan dari pembuatan aplikasi media sosial dengan menggunakan kompresi LZ77 dan Huffman berbasis *android*. Pada langkah selanjutnya butuh adanya perancangan tampilan aplikasi agar mudah dimengerti oleh pengguna sesuai kebutuhan dan fungsi perangkat lunak tersebut.

c. Pengkodean.

Pada tahap ini, dibutuhkan penerjemah dari tahap desain sistem ke dalam suatu bahasa pemrograman. Bahasa pemrograman yang digunakan untuk menyelesaikan aplikasi media sosial ini adalah bahasa pemrograman *javascript* dan untuk *database* memakai layanan lokal yaitu MySQL.

d. Pengujian.

Setelah pada tahap pembuatan program dan algoritma selesai, maka ditahap selanjutnya adalah pengujian aplikasi yang sudah dibuat.

e. Penerapan dan *Maintenance*.

Tahapan ini berlaku jika sistem sudah lulus dalam tahap pengujian dan siap untuk dioperasikan. Biasanya, dilakukan juga proses *monitoring*, perubahan dan evaluasi jika diperlukan.

2.1 Algoritma Huffman

Algoritma ini termasuk dalam metode kompres statistic yang memanfaatkan perhitungan statistika (*Statistical Methods*) untuk melihat probabilitas kemunculan data dari sebuah *file* atau gambar. Tahapan proses kompresi algoritma Huffman antara lain :

- 1) Menghitung banyaknya jenis karakter dan jumlah dari masing masing karakter yang terdapat dalam sebuah *file*.
- 2) Menyusun setiap jenis karakter dengan urutan jenis karakter yang jumlahnya paling sedikit ke yang jumlahnya paling banyak.
- 3) Membuat pohon biner berdasarkan berdasarkan urutan karakter dari yang jumlahnya terkecil ke yang terbesar, dan member kode untuk tiap karakter.
- 4) Mengganti data yang ada dengan kode bit berdasarkan pohon biner.

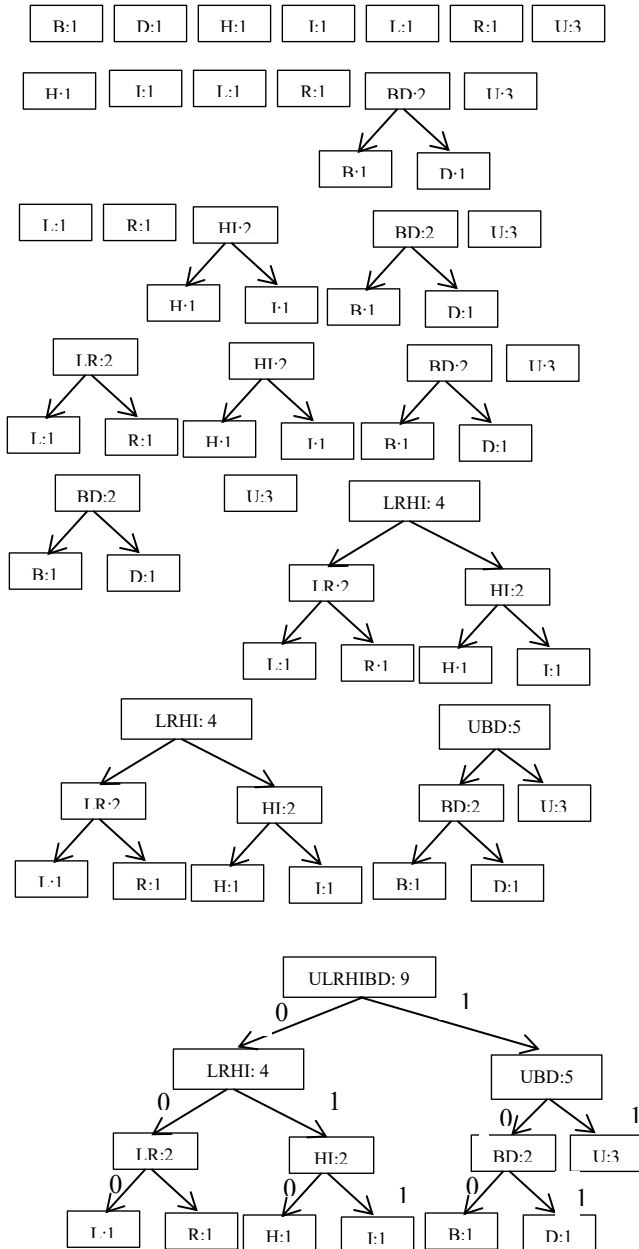
Menyimpan jumlah bit untuk kode bit yang terbesar, jenis karakter yang diurutkan dari frekuensi keluarnya terbesar ke terkecil beserta data yang sudah berubah menjadi kode bit sebagai data hasil kompresi.[3]. Langkah-langkah pembentukan pohon Huffman adalah sebagai berikut :

- 1) Pertama, membaca semua karakter di dalam teks untuk menghitung frekuensi kemunculan setiap karakter. Karakter penyusun teks dinyatakan sebagai pohon bersimpul tunggal. Setiap simpul di-*assign* dengan frekuensi kemunculan karakter tersebut.
- 2) Menerapkan algoritma greedy dengan cara menggabungkan dua buah pohon yang mempunyai frekuensi kemunculan terkecil pada sebuah akar. Setelah digabungkan akar tersebut akan mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon-pohon penyusunnya.
- 3) Ulangi langkah ke 2 sampai hanya tersisa satu buah pohon Huffman. Agar proses pemilihan dua pohon yang akan digabungkan berlangsung dengan cepat, maka memiliki urutan menaik berdasarkan frekuensi kemunculannya. [3].

Sebagai contoh, dalam kode ASCII huruf "BUDILUHUR" memiliki panjang 9 karakter membutuhkan representasi $9 \times 8 \text{ bit} = 72 \text{ bit}$ (9 *Byte*), dengan rincian sebagai berikut :

B = 01000010 U = 01010101
 U = 01010101 H = 01001000
 D = 01000100 U = 01010101
 I = 01001001 R = 01010010 L =
 01001100

Pada *string* di atas, frekuensi kemunculan U=3,
 B=1, D=1, H=1, I=1, L=1, R=1.



Gambar 1 : Pohon Huffman untuk Karakter “BUDILUHUR”[8].

Table 1 : Proses Encoding Algoritma Huffman.

Karakter	Muncul	Biner	Panjang	hitung
B	1	100	3	1 x 3 = 3
U	3	11	2	3 x 2 = 6
D	1	101	3	1 x 3 = 3
I	1	011	3	1 x 3 = 3

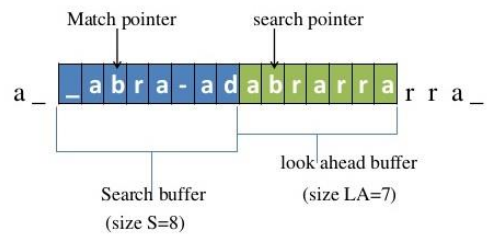
L	1	000	3	1 x 3 = 3
H	1	010	3	1 x 3 = 3
R	1	001	3	1 x 3 = 3
			Total =	24bit

$$\begin{matrix} B & U & D & I & L & U & H & U & R \\ \wedge & \wedge & \wedge & \wedge & \wedge & \wedge & \wedge & \wedge & \wedge \\ 100111010110001101011001 \end{matrix}$$

Gambar 2 : Proses Decoding Algoritma Huffman.

2.2 Algoritma LZ77

Algoritma LZ77 disebut juga jendela berjalan karena proses kerjanya. Prinsip dari algoritma ini adalah menggunakan sebagian input karakter yang telah dikodekan sebelumnya sebagai *dictionary* (kamus). Bagian *dictionary* ini ibaratkan sebuah jendela yang dapat digeser dari kiri ke kanan. Jendela ini secara dinamis merupakan kamus untuk mencari symbol input dengan pola tertentu. Ketika jendela ini bergerak dari kiri ke kanan, isi dari kamus dan input karakter yang dicari polanya juga akan berubah. Jendela ini dibagi menjadi dua bagian, bagian pertama disebut *search buffer* (*history buffer*), yang merupakan input karakter yang sudah dikodekan. Jendela kedua disebut *lookahead buffer*, yang berisi input karakter yang dikodekan namun hanya separuh bagian yang dikodekan. Ukuran dari masing-masing *buffer* ini telah ditentukan sebelumnya, dalam implementasinya nanti, *search buffer* akan memiliki panjang beberapa ribu byte, dan *lookahead buffer* panjangnya hanya puluhan byte.[1].



Gambar 3 : Gambaran Algoritma LZ77.

Dalam melakukan proses algoritma kompresi LZ77, ada beberapa tahapan yang harus dilakukan terlebih dahulu. Langkah pertama harus menentukan terlebih dahulu besar *history buffer*(*search buffer*) dan besar *lookahead buffer*. Langkah kedua isikan barisan karakter inputan ke dalam *lookahead buffer*. Apabila *history buffer* masih kosong atau tidak terdapat karakter pola yang sama di dalam *lookahead buffer* dibandingkan dengan *history buffer*, maka ditentukan nilai *token* dengan format *offset* bernilai 0, *length match*

bernilai 0, lalu diikuti 1 karakter *missmatch* di awal *lookahead buffer*. Namun apabila ditemukan pola maka akan ada *token* dengan format *offset* bernilai jarak dari *history buffer* sampai ditemukannya pola, *length match* bernilai panjang pola yang ditemukan sama, dan karakter *missmatch* diisi 1 karakter diluar dari pola yang ditemukan. Kemudian potong karakter yang sudah diubah menjadi *token* tersebut dari *lookahead buffer* dan diisikan ke dalam *history buffer*, yang menandakan karakter tersebut telah diproses kompresi. Isi *lookahead buffer* hingga penuh dari barisan inputan yang tersisa, dan lakukan proses pencocokan pola karakter seperti tadi hingga ditemukan *end of file*. Apabila sudah ditemukan *end of file* dan *lookahead buffer* sudah tidak terdapat karakter tersisa, maka akan didapatkan token hasil kompresi LZ77 [2].

Table 2 : Proses Encoding Algoritma LZ77.

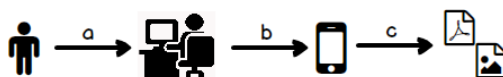
History Buffer	Lookahead Buffer	Token
	AKUADALAHAKU	(0,0,A)
A	KUADALAHAKU	(0,0,K)
AK	UADALAHAKU	(0,0,U)
AKU	ADALAHAKU	(3,1,D)
AKUAD	ALAHAKU	(2,1,L)
AKUADAL	AHAKU	(2,1,H)
AKUADALAH	AKU	(9,3,null)

Hasil kompresi dapat dilihat pada kolom nomer 3 pada bagian *token* yaitu : A,K,U,D,L dan H, null diartikan sudah tidak ada lagi kata atau sudah menjadi *end of file* pada *lookhead buffer*.

Table 2 : Proses Decoding Algoritma LZ77.

input		9	8	7	6	5	4	3	2	1
(0,0,A)										A
(0,0,K)									A	K
(0,0,U)								A	K	U
(3,1,D)						A	K	U	A	D
(2,1,L)				A	K	U	A	D	A	L
(2,1,H)		A	K	U	A	D	A	L	A	H
(9,3,nu ll)	AK U	A	D	A	L	A	H	A	K	U

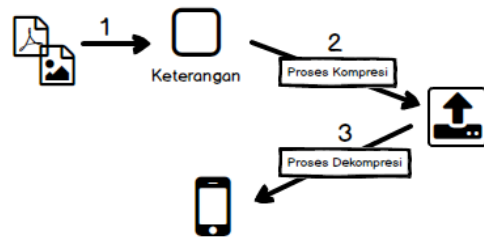
3. RANCANGAN SISTEM DAN APLIKASI



Gambar 4: Skema Garis Besar Aplikasi.

Secara garis besar aplikasi dibagi menjadi 3 poin penting yaitu :

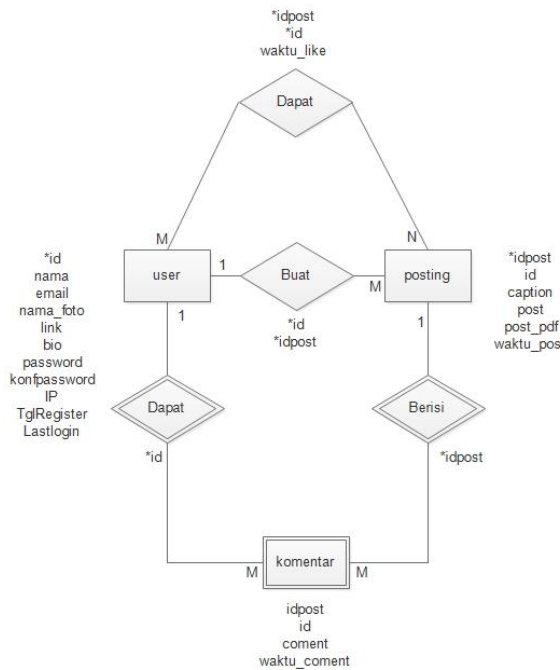
- a) Mahasiswa atau warga kampus yang ingin menggunakan aplikasi, namun belum memiliki akun sosial media ini harus melakukan proses register terlebih dahulu, namun jika sudah memiliki akun yang terdaftar bisa langsung melakukan proses login.
- b) setelah proses register berhasil, warga kampus akan diarahkan ke halaman utama aplikasi. Pada halaman utama aplikasi dibagi 3 menu yaitu :
 - (1) Menu Home
 - (2) Menu Posting
 - (3) Menu Profile
- c) Pada bagian Menu Posting pengguna dapat membagikan file gambar atau file PDF (*Portable Document Format*) dengan memberikan keterangan tentang file gambar atau PDF tersebut sebelum membagikan ke *public*.



Gambar 5 : Skema Proses Membagikan File.

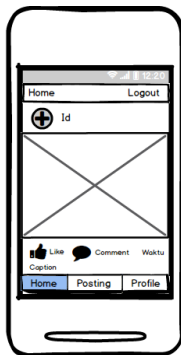
Terdapat 3 poin penting dalam proses membagikan file yaitu :

- (1) Warga kampus bisa memilih file gambar (PNG,JPEG,BMP) atau file PDF atau menggunakan kamera langsung. Kemudian, warga kampus akan menulis keterangan yang berhubungan dengan file yang dipilih, kemudian klik tombol bagikan.
- (2) Proses berikutnya adalah melakukan kompresi terhadap file yang dikirim, dengan menggunakan algoritma kompresi LZ77 terlebih dahulu, kemudian ditambahkan dengan algoritma kompresi Huffman. Setelah selesai proses kompresi, *file* akan dikirim ke sisi *server* dan di proses berserta data yang ada untuk di upload dan disimpan di server dan *database* MYSQL.
- (3) *File* yang tersimpan di *server* akan dibaca oleh program dan di dekomposisi menggunakan algoritma Huffman ditambahkan dengan algoritma LZ77, setelah file selesai di dekomposisi maka akan di tampilkan di menu Home dan di menu Profile.



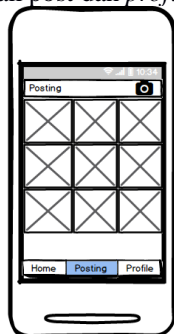
Gambar 6: Gambar Entity Relationship Diagram.

3.1 Rancangan Sistem



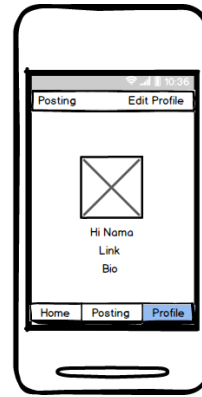
Gambar 7: Rancangan Sistem Menu Home.

Rancangan sistem menu home berguna untuk pengguna melihat postingan yang telah dikirim oleh pengguna lain, disini pengguna dapat memberikan Like, Comment, detail post dan profile pengirim..



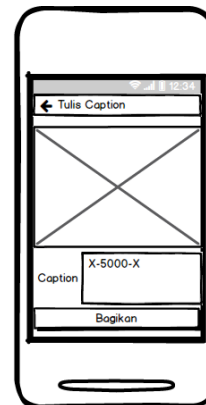
Gambar 8: Rancangan Sistem Menu Posting.

Rancangan sistem menu posting berguna untuk melihat daftar postingan yang telah dikirim oleh pengguna dan juga untuk proses pengiriman postingan dengan menekan icon kamera.



Gambar 9: Rancangan Sistem Menu Profile.

Rancangan sistem menu profile berguna untuk melihat profile dari pengguna, mengganti foto profile, dan data profile pengguna.



Gambar 10: Rancangan Sistem Tulis Caption.

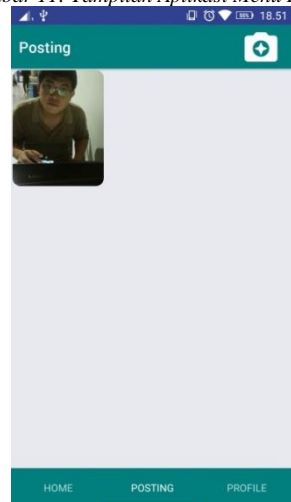
Pada rancangan sistem tulis caption ini pengguna yang sudah memilih gambar atau file pdf yang akan dikirim dapat melihat preview terlebih dahulu dan dapat menulis caption yang menceritakan tentang postingan tersebut sebelum dibagikan. Pada proses bagikan inilah proses kompresi LZ77 dan Huffman dilakukan sebelum ditaruh di server dan di dekompresi lagi sebelum data di tampilkan.

4. HASIL DAN PEMBAHASAN.

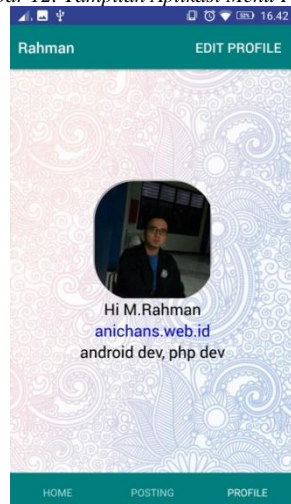
Pada bagian hasil dan pembahasan ini akan ditampilkan contoh bagian tampilan program dari aplikasi yang dikembangkan dan juga ada terdapat grafik perbandingan algoritma LZ77, Huffman, dan penggabungan LZ77 dan Huffman berdasarkan jenis file yang diproses.



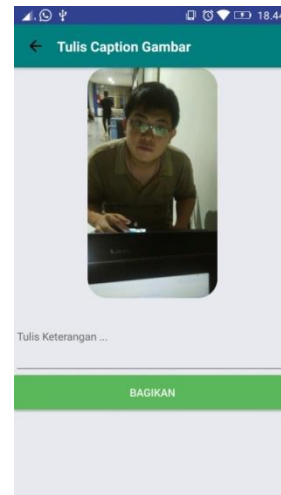
Gambar 11: Tampilan Aplikasi Menu Home.



Gambar 12: Tampilan Aplikasi Menu Posting.



Gambar 13: Tampilan Aplikasi Menu Profile.

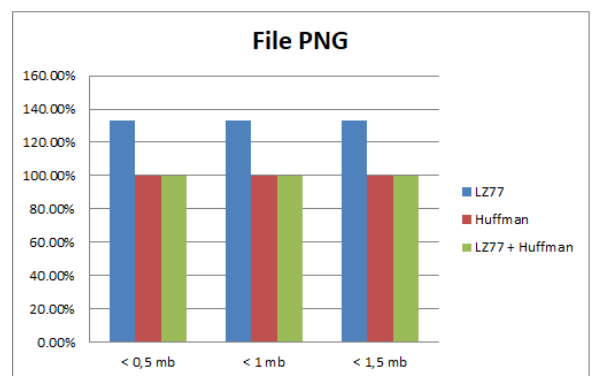


Gambar 14: Tampilan Aplikasi Tulis Caption.



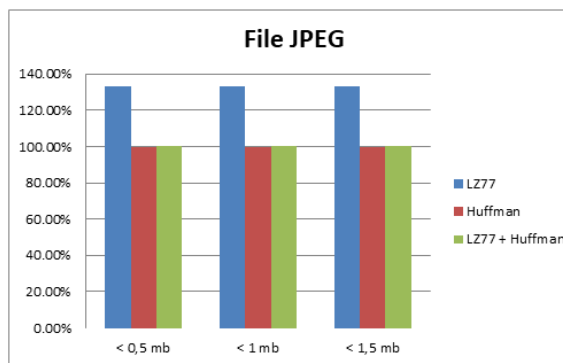
Gambar 15: Grafik Rasio File PDF.

Jika diperhatikan dari grafik rasio file PDF dapat ditarik kesimpulan bahwa algoritma LZ77 + Huffman memiliki hasil rasio kompresi yang lebih kecil dibandingkan dengan LZ77 atau Huffman jika berdiri sendiri.



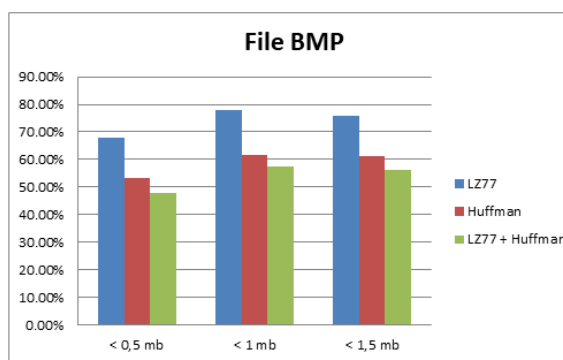
Gambar 16: Grafik Rasio File PNG.

Jika diperhatikan dari grafik rasio file PNG dapat ditarik kesimpulan bahwa algoritma Huffman jika berdiri sendiri dibandingkan dengan LZ77 + Huffman tidak memiliki perbedaan dalam hal rasio.



Gambar 17: Grafik Rasio File JPEG.

Jika diperhatikan dari grafik rasio file JPEG dapat ditarik kesimpulan bahwa algoritma Huffman jika berdiri sendiri dibandingkan dengan LZ77 + Huffman tidak memiliki perbedaan dalam hal rasio, hanya algoritma LZ77 saja yang tingkat rasio yang tinggi.



Gambar 18: Grafik Rasio File BMP.

Jika diperhatikan dari grafik rasio file BMP dapat ditarik kesimpulan bahwa algoritma LZ77 + Huffman memiliki hasil rasio kompresi yang lebih kecil dibandingkan dengan LZ77 atau Huffman jika berdiri sendiri.

Rasio yang diatas 100 % memiliki arti bahwa pada saat kompresi dilakukan, file yang terkompresi memiliki size atau ukuran file yang membengkak atau lebih besar dari pada file aslinya. Itu bisa terjadi dikarenakan algoritma yang digunakan yaitu LZ77 dan Huffman merupakan jenis kompresi Lossless yang biasa digunakan pada text, bukan file ataupun gambar.

5. KESIMPULAN DAN SARAN

- Aplikasi ini dapat mengirim file PNG, JPEG, BMP dan PDF yang dapat digunakan pengguna, khususnya pada warga kampus untuk memberikan informasi kepada mahasiswa atau warga kampus lainnya tentang event atau informasi apapun yang berhubungan dengan Universitas Budi Luhur dengan menerapkan algoritma kompresi LZ77 dan Huffman.

- Pada aplikasi ini warga kampus yang menggunakan aplikasi dapat melakukan sistem like, comment, mengedit profile atau foto profile, dan juga dapat melihat profile dari pengguna lain.
- Metode LZ77 dan Huffman diterapkan di aplikasi pada saat warga kampus ingin membagikan postingan. Proses pertama dilakukan dengan kompresi algoritma LZ77 jika proses sudah selesai maka dilanjutkan dengan proses algoritma kompresi Huffman setelah itu data di simpan di database dan server. Kemudian dilakukan proses dekomposisi menggunakan algoritma dekomposisi Huffman dan setelah selesai dilanjutkan dengan algoritma dekomposisi LZ77.
- Jika dilihat hasil rasio dari tabel pengujian dapat ditarik kesimpulan bahwa algoritma LZ77 memiliki kecepatan kompresi yang lebih baik dari pada algoritma Huffman, namun kualitas kompresi pada algoritma Huffman lebih baik dari pada algoritma LZ77.
- Penggabungan algoritma LZ77 dan algoritma Huffman memiliki kecepatan yang berada di tengah-tengah diantara algoritma LZ77 dan algoritma Huffman, namun jika dilihat dari segi kualitas kompresi penggabungan algoritma LZ77 dan algoritma Huffman menghasilkan rasio kompresi yang lebih kecil dibandingkan dengan algoritma LZ77 berdiri sendiri dan algoritma Huffman berdiri sendiri.
- Algoritma LZ77 dirasa kurang cocok untuk file PNG, JPEG dan PDF jika berdiri sendiri. Karena, file yang dikompresi menjadi lebih besar ukuran filenya.
- Algoritma Huffman dirasa kurang cocok untuk file PNG, JPEG dan PDF jika berdiri sendiri. Karena, file yang dikompresi menjadi hanya berkurang sedikit ukuran filenya.

Terdapat pula saran-saran yang bisa dijadikan pertimbangan dalam pengembangan aplikasi, antara lain:

- Aplikasi dapat ditingkatkan, tidak hanya dapat kompresi dan dekomposisi file gambar dan PDF namun, juga dapat kompresi dan dekomposisi file video maupun audio.
- Fitur yang dapat ditambahkan adalah push-notification agar pengguna dapat mengetahui postingan terbaru.
- Selain teknik kompresi Lossless, aplikasi dapat ditingkatkan dengan menerapkan algoritma kompresi Lossy seperti DCT, JPEG dll, dikarenakan untuk algoritma LZ77 dan Huffman yang menerapkan algoritma Lossless yang dibuat khusus untuk kompresi text, dirasa kurang cocok untuk file gambar ataupun PDF.
- Fitur yang dapat ditambahkan adalah verifikasi akun, untuk membuktikan apakah benar akun tersebut milik pengguna.

- e. Fitur yang dapat ditambahkan adalah sistem pencarian pengguna lain.

6. DAFTAR PUSTAKA

- [1] Siswanto *et al.* (2016) 'Pengamanan Data Dengan Menggunakan Algoritma Kriptografi Aes , Rc4 Dan Kompresi Lz77', *Seminar Nasional Telekomunikasi dan Informatika (SELESIK 2016)*, 1, pp. 115–120.
- [2] Nugraha, C. P., Santosa, R. G. and A.A., L. C. (2014) 'Perbandingan Metode LZ77, Metode Huffman dan Metode Deflate Terhadap Kompresi Data Teks', *Jurnal Informatika*, 10(2), pp. 80–91.
- [3] Yana, G. I. and Hondro, R. K. (2016) 'Implementasi Algoritma Huffman Dan Lz78 Untuk Kompresi Data', *Jurnal Riset Komputer (JURIKOM)*, 3(6), pp. 42–44.