

Optimasi Web Service dengan Penerapan Algoritma Kompresi LZW: Studi Kasus Aplikasi BluCampus Universitas Budi Luhur

Mochammad Andika Putra¹⁾, Achmad Solichin²⁾

¹⁾Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

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

E-mail : mochammadandika9@gmail.com¹⁾, achmad.solichin@budiluhur.ac.id²⁾

Abstrak

REST web service semakin populer digunakan sebagai media pertukaran data pada saat ini. Hal ini disebabkan oleh meningkatnya jumlah aplikasi mobile yang kian pesat. Dengan menggunakan REST web service, pertukaran data antara mobile dengan server menjadi lebih mudah, karena web service dapat dibangun dengan menggunakan bahasa pemrograman yang berbeda dengan aplikasi pengguna yang menggunakan layanan sistem. Tetapi, seiring terjadinya pertukaran data menyebabkan bertambahnya kapasitas ruang penyimpanan pada database server. Oleh karena itu pada penelitian ini diimplementasikan web service berbasis REST sebagai media pertukaran data dan algoritma kompresi LZW untuk mengurangi ruang penyimpanan pada database server. Pada penelitian ini dianalisis seluruh layanan web service yang akan digunakan pada aplikasi android, beserta kelebihan dan kekurangan algoritma kompresi pada data teks yang berjumlah 100 hingga 1600 karakter. Hasil penelitian menunjukkan seluruh layanan web service yang dibuat dapat diimplementasikan pada aplikasi android dengan baik. Untuk kompresi sendiri semakin banyaknya jumlah data maka tingkat persentase penghematan akan semakin baik yaitu mencapai angka 51%, tetapi semakin banyaknya jumlah data yang akan dikompresi, response time yang diberikan oleh server juga akan meningkat mencapai 260 milliseconds.

Kata kunci: REST, RESTful, Web Service, Algoritma LZW, Kompresi data

1. PENDAHULUAN

Perkembangan aplikasi *mobile* pada saat ini berkembang sangat pesat, hal ini dikarenakan semakin mudahnya mendapatkan perangkat *smartphone* dengan harga yang terjangkau. Oleh karena itu, banyak dikembangkan aplikasi berbagai jenis yang dibuat menggunakan berbagai macam bahasa pemrograman yang dapat diunduh oleh pengguna *smartphone* sesuai kebutuhan. Salah satu jenis aplikasi *smartphone* yang banyak digunakan adalah aplikasi yang bisa memberikan informasi kepada penggunanya.

Aplikasi *smartphone* dengan fitur informasi promosi dan informasi sudah pasti membutuhkan data yang dinamis seiring dengan berubahnya data terkait [5]. Solusi yang diajukan adalah menyimpan data di satu *database server* [2]. Selanjutnya setiap aplikasi melakukan *request* untuk mendapatkan data dari sebuah *server*. Begitu juga pada saat pengiriman data, setiap aplikasi akan mengirimkan data ke *server*.

Untuk mendukung *scalability* maka digunakan teknologi RESTful *web service* sebagai solusi pertukaran data [5]. Dengan memanfaatkan RESTful *web service* maka setiap *platform* perangkat *mobile* seperti ponsel berbasis Android, tablet berbasis windows, aplikasi web, maupun aplikasi desktop yang menggunakan data internet, dapat mengakses layanan data dari *server*, kapanpun dan dimanapun selama terhubung dengan jaringan internet.

Namun, seiring bertambahnya pengguna aplikasi yang mengirimkan data ke *server*, maka akan berdampak dalam hal meningkatnya ukuran penyimpanan pada *database server* tersebut. Untuk mengatasi hal tersebut maka setiap data yang dikirimkan dapat diperkecil ukurannya tanpa

menghilangkan makna informasi di dalamnya, dengan menggunakan algoritma kompresi.

Pada penelitian ini diimplementasikan RESTful *web service* dalam hal pertukaran data. Sedangkan untuk proses kompresi dan dekompresi akan menggunakan algoritma LZW. Hal ini dikarenakan algoritma kompresi LZW yang merupakan *dictionary based compression* sangat efektif mengkompresi data teks yang memiliki banyak pola huruf, kata ataupun kalimat yang berulang [3]. Semakin banyak pembendaharaan gabungan *string* dalam dictionary tambahan pada suatu *plaintext* maka semakin baik hasil kompresi *plaintext* tersebut [2]. Tujuan dari implementasi ini adalah untuk memperkecil ukuran ruang penyimpanan pada *database server* dengan algoritma kompresi LZW.

2. TINJAUAN PUSTAKA

2.1. Web Service

Web service adalah standar yang digunakan untuk melakukan pertukaran data antar aplikasi atau sistem, karena aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda [4]. Contoh metode dari *web service* antara lain adalah SOAP dan REST. *Web service* yang berbasis arsitektur REST kemudian dikenal sebagai RESTful *web service* [1]. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST.

Teknologi *web service* sudah banyak digunakan dalam berbagai penelitian, terutama untuk mengatasi aksesibilitas data dari berbagai platform. Penelitian oleh Aziz dkk [1] memanfaatkan *web service Moodle*

berbasis REST-JSON untuk membangun Moodle Online Learning Extension berbasis Android. Sementara itu, Jumardi dan Solichin [6] juga mengimplementasikan teknologi web service dalam aplikasi pengaduan masyarakat berbasis Android. Selain REST, metode SOAP juga masih sering digunakan dalam implementasi web service seperti pada penelitian oleh Surendra [5].

2.2. Kompresi

Kompresi adalah suatu teknik pemampatan data sehingga diperoleh file dengan ukuran yang lebih kecil daripada ukuran aslinya [2]. Kompresi bekerja dengan mencari pola-pola perulangan pada data dan menggantinya dengan sebuah penanda tertentu [3]. Pemampatan data umumnya diterapkan pada mesin komputer, hal ini dilakukan karena setiap simbol yang dimunculkan pada komputer memiliki nilai bit-bit yang berbeda. Misal pada ASCII setiap simbol yang dimunculkan memiliki panjang bit 8 bit, misal kode A pada ASCII mempunyai nilai desimal = 65, jika dirubah dalam bilangan biner menjadi 01000001. Pemampatan data digunakan untuk mengurangi jumlah bit-bit yang dihasilkan dari setiap simbol yang muncul [2]. Dengan pemampatan ini diharapkan dapat mengurangi (memperkecil ukuran data) dalam ruang penyimpanan.

2.3. Rasio Kompresi

Rasio Kompresi merupakan ukuran persentase data yang telah berhasil dimampatkan [3]. Secara matematis rasio pemampatan data dituliskan sebagai berikut.

$$\text{Rasio} = 100\% - \frac{\text{Ukuran Data Terkompresi} \times 100\%}{\text{Ukuran Data Asli}} \quad (2.1)$$

3. METODE PENELITIAN

3.1. Metode Penelitian

Untuk dapat mengimplementasikan sistem di atas, maka secara garis besar metode penelitian yang akan digunakan adalah sebagai berikut :

- Studi literatur dan teori penunjang untuk memperoleh informasi dengan mempelajari buku-buku literatur atau karya lainnya yang membahas mengenai RESTful web service beserta algoritma LZW atau untuk menunjang pembuatan perangkat lunak yang berhubungan dengan materi penulisan tugas akhir.
- Penerapan metode RESTful web service dan algoritma LZW dalam perancangan sistem.
- Analisa permasalahan untuk mengetahui dan menentukan algoritma yang digunakan sehingga dapat menentukan cara yang paling efektif dalam penyelesaian suatu permasalahan dalam proses kompresi dan dekompresi.

- Evaluasi program dengan melakukan pengujian dan pengoperasian sistem secara keseluruhan evaluasi program dan pengujian pada suatu sistem sangat diperlukan untuk mengetahui kestabilan sistem yang telah dibuat.

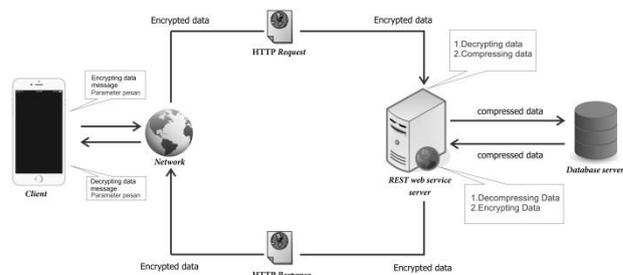
3.2. Analisis dan Penyelesaian Masalah

Pada saat ini kemajuan aplikasi berbasis mobile terus meningkat pesat. Beraneka ragam aplikasi tersebut dibuat menggunakan bahasa pemrograman yang beraneka ragam. Keadaan tersebut dapat menimbulkan masalah dalam proses pertukaran data antar perangkat yang menggunakan aplikasi dan platform yang berbeda. Pertukaran data yang dimaksud salah satunya ketika pengguna aplikasi mengirim sebuah data yang akan di simpan dalam suatu penyimpanan yang biasa disebut database, akan berdampak pada meningkatnya jumlah data yang tersimpan sehingga menambah kapasitas dari database seiring bertambahnya pengguna dari aplikasi tersebut.

Dari permasalahan yang telah diuraikan diatas, maka diperlukan adanya sebuah sistem yang dapat melakukan pertukaran data pada sebuah aplikasi tanpa memikirkan bahasa pemrograman apa yang dibuat pada aplikasi tersebut, sekaligus mengurangi ukuran data yang akan tersimpan pada *database*, sehingga kapasitas dari *database* diharapkan akan berkurang.

Untuk mengimplementasikan sistem tersebut maka akan digunakan sebuah arsitektur web services sebagai media untuk pertukaran data yang disebut REST *web service*, dan juga algoritma kompresi dengan tujuan untuk mengurangi ukuran data sehingga akan berdampak pada kapasitas ruang penyimpanan. Dalam proses pengiriman data dibutuhkan algoritma kompresi agar ukuran data tersebut dapat berkurang, kemudian pada saat pengembalian data akan dilakukan proses dekompresi agar data tersebut bisa ditampilkan pada aplikasi sesuai dengan yang aslinya.

Dengan adanya sistem ini, diharapkan dapat mengurangi ukuran kapasitas penyimpanan data pada *database*, dan juga terbentuknya aplikasi yang dapat berjalan dengan baik.

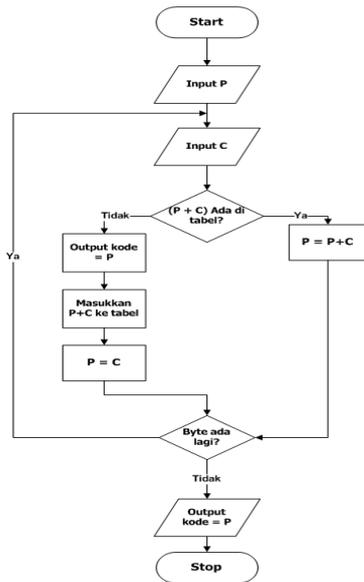


Gambar 1. Alur Kerja Kompresi dan Dekompresi

3.3. Flowchart dan Algoritma LZW

Pada bagian ini akan dijelaskan flowchart dan algoritma dari metode LZW yang digunakan pada

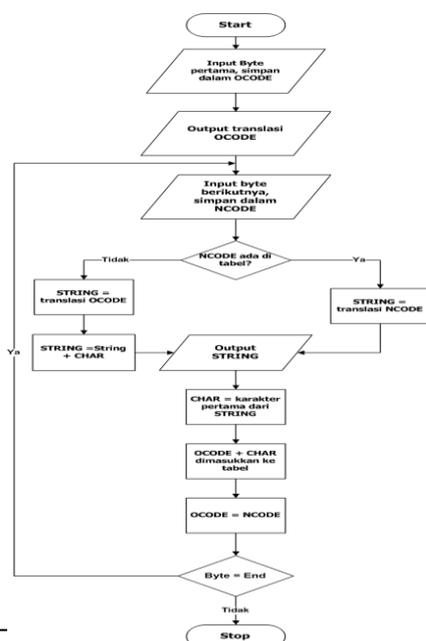
saat data pesan disimpan maupun dikembalikan oleh *web service*.



Gambar 2. Flowchart Proses Kompresi LZW

Algoritma dari *flowchart* kompresi LZW :

1. Start
2. Input byte pertama, simpan dalam P
3. Input byte berikutnya, simpan dalam C
4. If P+C ada ditabel then
5. P = P+C
6. Else
7. Output code = P
8. Masukkan P +C ke tabel
9. P = C
10. End if
11. If ada byte lagi?
12. Kembali ke baris 3
13. Else
14. Output code = P
15. End if



Gambar 3. Flowchart Proses Dekompresi LZW
Algoritma dari *flowchart* dekompresi LZW :

1. Start
2. Input byte pertama, simpan dalam OCODE
3. Output translasi OCODE
4. Input byte berikutnya, simpan dalam NCODE
5. If NCODE ada ditabel then
6. STRING = translasi NCODE
7. Else
8. STRING = translasi OCODE
9. STRING = STRING + CHAR
10. End if
11. Output STRING
12. CHAR = karakter pertama dari STRING
13. OCODE + CHAR dimasukkan ke tabel
14. OCODE = NCODE
15. If ada byte lagi?
16. Kembali ke baris 4
17. Else
18. End if

3.4. Rancangan Layanan Web Service

Bagian ini berisi berbagai layanan yang akan digunakan pada sistem blucareer, beserta penjelasan dari masing-masing layanan seperti nama layanan, nama fungsi, parameter, dan keluaran.

Tabel 1. Daftar Layanan Web Service

Nama Layanan	Fungsi	Path	Parameter
Data Kelas	GET	kelas	
Join Kelas	POST	kelas/{kelas}/join	Path parameter: id_kelas
Unjoin Kelas	POST	kelas/{kelas}/unjoin	Path parameter: id_kelas
Cari Kelas	POST	kelas/cari	Request body parameter : carikelas
Hapus Kelas	DELETE	kelas/{kelas}	Path parameter: id_kelas
Buat Kelas	POST	kelas	Request body parameter : nama_kelas, kursus, lokasi, jumlah, price, tanggal_acara, batas_tanggal_a cara, start, finish, tags
Ubah Kelas	PATCH	kelas/{kelas}	Path parameter: id_kelas Request body parameter : nama_kelas, kursus, lokasi, jumlah, price, tanggal_acara, batas_tanggal_a cara, start, finish, tags, status_kelas

Absensi Kelas	PATCH	kelas/{kelas}/updateuser	Path parameter: id_kelas Request body parameter : Id_user, status
Data Profile User	GET	user/profile	
Daftar followers user	GET	followers	
Daftar following user	GET	following	
Tambah skill user	POST	user/postskill	Request body parameter : skill
Tambah about me	POST	user/postaboutme	Request body parameter : about_me
Cari data user berdasarkan event	POST	cari	Request body parameter : Cari
Cari data user berdasarkan nama	POST	cariuser	Request body parameter : Cari
Cari data user berdasarkan skill	POST	cariskill	Request body parameter : cari
Follow user	POST	user/{user}/Follow	Path parameter: id_user
Unfollow user	POST	user/{user}/Unfollow	Path parameter: id_user
Data percakapan user	GET	/message	
Buat pesan	POST	sendmessagetoreceiver/{receiverId}	Path parameter: id_user Request body parameter : message

3.5. Spesifikasi Basis Data

Berikut ini adalah spesifikasi basis data secara detail tentang masing-masing basis data untuk menyimpan dan mengakses data yang digunakan pada layanan *web service*.

Tabel 2. Otorisasi

Field	Type	Length	Keterangan
id	int	10	id otorisasi
nama_otorisasi	varchar	30	nama_otorisasi
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah

Tabel 3. Status

Field	Type	Length	Keterangan
id	int	10	id status
nama_status	varchar	30	nama_status
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah

Tabel 4. Users

Field	Type	Length	Keterangan
id	int	10	id user
username	varchar	30	username user
nama	varchar	50	nama user
email	varchar	50	email user
password	varchar	50	password user
tanggal_lahir	date	10	tanggal lahir user
jenis_kelamin	enum	p,l	jenis kelamin user
alamat	text	255	alamat user
foto	varchar	191	foto user
about_me	varchar	191	tentang user
token	varchar	191	token user
status_aktif	tinyint	1	status aktif user
telepon	varchar	12	telepon user
ni	varchar	10	nomor induk user
id_status	int	10	status user
id_otorisasi	int	10	otorisasi user
remember_token	varchar	191	ingat token user
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah

Tabel 5. Skills

Field	Type	Length	Keterangan
id	int	10	id skill
skill	varchar	30	nama skill
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah
id_user	Int	10	id user

Tabel 6. Kelas

Field	Type	Length	Keterangan
id	int	10	id kelas
nama_kelas	varchar	50	nama kelas
kursus	varchar	50	kursus kelas
lokasi	varchar	255	lokasi kelas
jumlah	int	10	jumlah pendaftar
price	int	10	biaya kelas
tanggal_acara	date	10	tanggal acara keals
batas_tanggal_acara	date	10	batas tanggal acara
start	varchar	4	waktu mulai
finish	varchar	4	waktu selesai
tags	varchar	100	tag kelas
status_kelas	tinyint	1	status kelas
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah
id_user	int	10	id user

Tabel 7. Friends

Field	Type	Length	Keterangan
id_user	int	10	id user
id_friend	int	10	id user
id_user_follow	tinyint	1	status user follow
id_friend_follow	tinyint	1	status friend follow

Tabel 8. Conversations

Field	Type	Length	Keterangan
id	int	10	id conversation
user_one	int	10	id user
user_two	int	10	id user
status	tinyint	1	status
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah

Tabel 9. Messages

Field	Type	Length	Keterangan
id	int	10	id message
message	text	191	pesan
is_seen	tinyint	1	telah dilihat
delete_from_sender	tinyint	1	hapus pengirim
delete_from_receiver	tinyint	1	hapus penerima
user_id	int	10	id user
conversation_id	int	10	id conversation
created_at	timestamp	20	tanggal buat
updated_at	timestamp	20	tanggal ubah

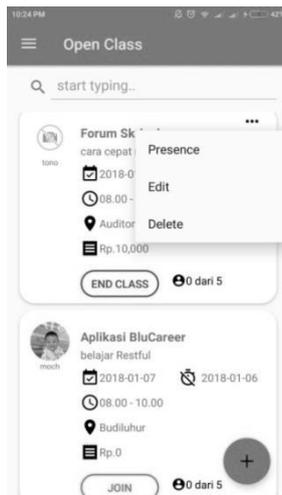
4. HASIL DAN PEMBAHASAN

4.1. Tampilan Layar

Bagian ini merupakan hasil dari implementasi layanan web service yang telah dikembangkan dalam bentuk aplikasi berbasis android, dan akan dijelaskan proses dari aplikasi tersebut mulai dari pertama kali dijalankan sampai selesai. Berikut mengenai tampilan-tampilan yang ada pada aplikasi ini.

4.1.1. Tampilan Menu *Open Class*

Pada tampilan ini menggunakan layanan Data kelas, untuk menampilkan data kelas yang sudah dibuat oleh *user*. Beserta layanan yang lain seperti Join kelas, Unjoin kelas, Cari kelas, Hapus kelas.



Gambar 4. Menu Open Class

4.1.2. Tampilan *Form Create Class*

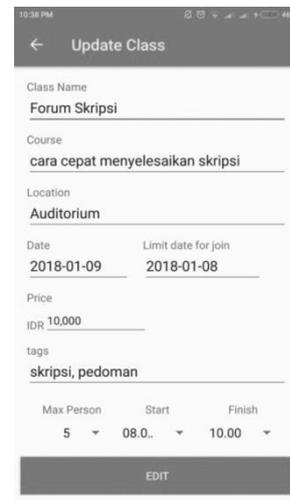
Pada tampilan ini menggunakan layanan Buat kelas, untuk mengirimkan inputan untuk membuat sebuah kelas.



Gambar 5. Form Create Class

4.1.3. Tampilan *Form Edit Class*

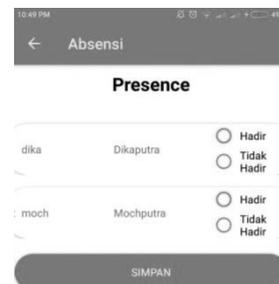
Pada tampilan ini menggunakan layanan Ubah kelas, untuk mengubah data kelas *user*.



Gambar 6. Form Edit Class

4.1.4. Tampilan *Form Presence*

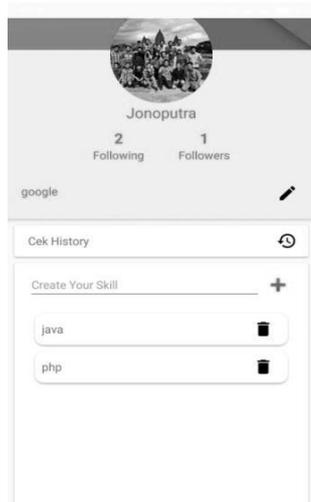
Pada tampilan ini menggunakan layanan Absensi kelas, untuk absen nama *user* dari kelas tersebut.



Gambar 7. Form Presence

4.1.5. Tampilan *Menu Profile*

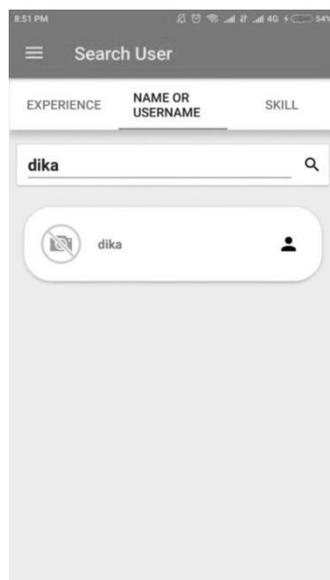
Pada tampilan ini terdapat layanan Data profile user, Daftar followers user, Daftar following user, Tambah skill user, Tambah about me user.



Gambar 8. Menu Profile

4.1.6. Tampilan Menu Search Friend

Pada tampilan ini terdapat layanan Cari data user berdasarkan event, nama, dan skill. Beserta follow, dan unfollow.



Gambar 9. Menu Search Friend

4.1.7. Tampilan Menu Message

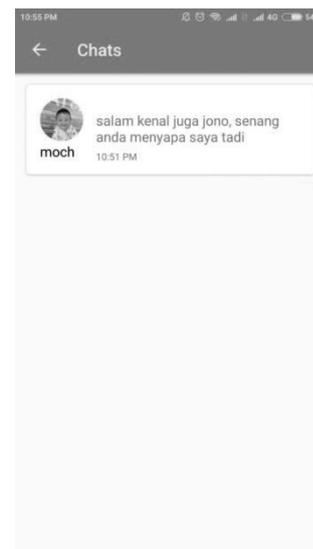
Pada tampilan ini menggunakan layanan Data percakapan *user*, untuk mendapatkan percakapan antar *user*, setelah melalui proses dekomposisi sebelumnya.



Gambar 10. Menu Message

4.1.8. Tampilan Form Message Detail

Pada tampilan ini terdapat layanan Buat pesan, untuk membuat suatu pesan untuk user lain. Selain itu terdapat proses kompresi beserta dekompresi.



Gambar 11. Form Message Detail

4.2. Uji Coba Program

Pada bagian ini, akan dilakukan pengujian terhadap layanan web service yang digunakan, dan proses kompresi dan dekomposisi untuk data pesan.

4.2.1. Uji Coba Layanan Web Service

Tabel dibawah ini merupakan hasil pengujian seluruh layanan *web service* yang sudah diimplementasikan pada aplikasi android.

Tabel 10. Layanan Web Service

No	Nama Layanan	Status
1	Data kelas	Berhasil
2	Join kelas	Berhasil
3	Unjoin kelas	Berhasil
4	Cari kelas	Berhasil
5	Hapus kelas	Berhasil
6	Buat kelas	Berhasil
7	Ubah kelas	Berhasil
8	Absensi kelas	Berhasil
9	Data profile user	Berhasil
10	Daftar followers user	Berhasil
11	Daftar following user	Berhasil
12	Tambah skill user	Berhasil
13	Tambah about me user	Berhasil
14	Cari data user berdasarkan event	Berhasil
15	Cari data user berdasarkan nama	Berhasil
16	Cari data user berdasarkan skill	Berhasil
17	Follow user	Berhasil
18	Unfollow user	Berhasil
19	Data percakapan user	Berhasil
20	Buat pesan	Berhasil

4.2.2. Uji Coba Kompresi dan Dekompresi Data Pesan

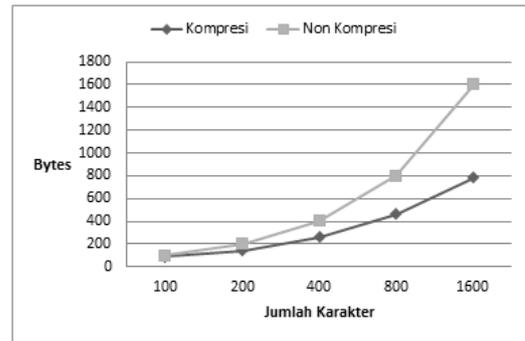
Pada Tabel di bawah ini adalah hasil pengujian dari kompresi maupun dekompresi pada sejumlah teks dengan jumlah karakter 100, 200, 400, 800, dan 1600 dengan perulangan 10 kali dengan bantuan tool *RESTful Stress* menggunakan internet berkecepatan up to 10 mbps.

Pengujian kompresi dan dekompresi LZW dilakukan dengan parameter pengujian Jumlah Karakter (JK), Ukuran Data Awal (UD) dalam ukuran bytes, Ukuran Data Hasil Kompresi (UDH) dalam ukuran bytes, *Compression Time* (CT) dalam ukuran milliseconds, *Compression Ratio* (CR) dalam ukuran percent, *Decompression Time* (DT) dalam ukuran milliseconds, dan Waktu Tanpa Menggunakan Kompresi (WK) dalam ukuran milliseconds [2].

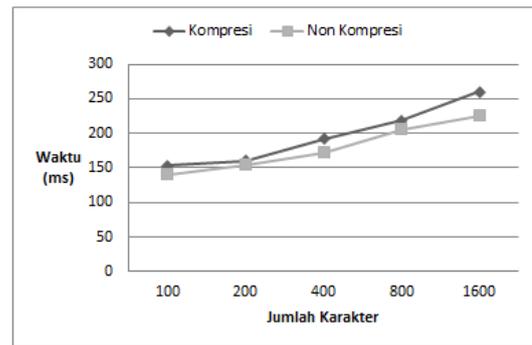
Tabel 11. Rata-rata hasil pengujian

No	JK	UKD	UKH	CT	CR	DT	WK
1	100	102	86	153	15,69	137	140
2	200	202	145	160	28,22	140	154
3	400	402	259	192	35,57	154	172
4	800	802	461	218	42,52	166	206
5	1600	783	260	260	51,12	206	225

Hasil dari tabel diatas menunjukkan perbedaan ukuran data antara sebelum dilakukan proses kompresi dengan data kompresi yang ditunjukkan pada kolom *Compression Ratio* (CR), beserta perbedaan waktu yang terjadi antara terdapat proses kompresi dengan tanpa kompresi.



Gambar 12. Grafik Perbandingan Data Hasil Kompresi dengan Tanpa Kompresi



Gambar 13. Grafik Perbandingan Waktu Kompresi dengan Tanpa Kompresi

4.3. Evaluasi Program

Setelah dilakukan analisa dan hasil pengujian aplikasi dapat ditemukan beberapa kelebihan dan kekurangan dari aplikasi ini, yaitu sebagai berikut.

Kelebihan Program :

- Aplikasi mudah dipahami dan digunakan oleh user.
- Semua layanan web service dapat berfungsi dengan baik.
- Data pesan dapat dikompresi sehingga dapat mengurangi ruang penyimpanan pada database.
- Hasil dari kompresi data pesan dapat kembali seperti aslinya.

Kekurangan Program :

- Pesan yang dibuat pada aplikasi ini hanya dapat berbentuk teks, belum mendukung fitur emoji.
- Dengan menggunakan teknik kompresi ini, response time yang diberikan server sedikit lebih lama jika dibandingkan dengan tidak menggunakan teknik kompresi.

5. KESIMPULAN

Berdasarkan perancangan, pembuatan, serangkaian uji coba, dan analisa program dari aplikasi ini, maka dapat diambil beberapa kesimpulan yaitu :

- Seluruh layanan web service yang dibuat dapat diimplementasikan pada aplikasi android dengan baik.
- Hasil dari proses kompresi untuk data akan semakin baik jika jumlah data besar, tetapi kurang baik ketika jumlah data sedikit.
- Waktu yang diperlukan untuk proses kompresi maupun dekompresi berbanding lurus dengan hasilnya, jika data semakin banyak maka waktu yang diberikan juga akan semakin meningkat.
- Hasil Response time server dengan menggunakan teknik kompresi lebih lama dibandingkan dengan tidak menggunakan kompresi.

6. DAFTAR PUSTAKA

- [1] A. Aziz, Wiharto, and B. Wicaksono, "Pemanfaatan Web Service Moodle Berbasis REST-JSON untuk Membangun Moodle Online Learning Extension berbasis Android," *J. Itsmart*, vol. 2, no. 2, pp. 1–6, 2013.
- [2] Hidayat, W. Zarman, and T. Pamungkas, "Implementasi Algoritma Kompresi Lzw Pada Database Server," *J. Ilm. Komput. dan Inform. (KOMPUTA)*, vol. 2, no. 1, pp. 7–13, 2013.
- [3] R. Maulunida and A. Solichin, "Optimization of LZW Compression Algorithm With Modification of Dictionary Formation," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 12, no. 1, p. 73, 2018.
- [4] M. A. Rahman, I. Kuswardayan, and R. R. Hariadi, "Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android," *J. Tek. POMITS*, vol. 2, no. 1, pp. 2–5, 2013.
- [5] M. R. S. Surendra, "Implementasi PHP Web Service Sebagai Penyedia Data Aplikasi Mobile," *J. Tek. Inform.*, vol. VI, no. 2, pp. 85–93, 2014.
- [6] Jumardi, A. and Solichin, A., 2016. PROTOTIPE APLIKASI LAYANAN PENGADUAN MASYARAKAT BERBASIS ANDROID DAN WEB SERVICE. *Telematika MKOM*, 8(1), pp.81-88.