

APLIKASI KOMPRESI CITRA DENGAN MENGGUNAKAN ALGORITMA LEMPERL ZIV WELCH (LZW)

Akbar Nur Alim, Haris Yuana, Filda Febrinita

Program Studi Teknik Informatika S1, Fakultas Teknologi Informasi
Universitas Islam Balitar, Sananwetan Kota Blitar, Indonesia
akbarnuralim8@gmail.com

ABSTRAK

Citra digital kini banyak digunakan di berbagai bidang. Mulai dari kebutuhan sehari-hari seperti percetakan foto, konten *website*, konten media sosial, dan citra satelit, secara tidak langsung meningkatkan kebutuhan penyimpanan data. Sehingga, dicari teknik yang dapat memperkecil ukuran *file* citra melalui kompresi. Kompresi adalah teknik untuk menghemat media penyimpanan yang digunakan. Kompresi menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan *bandwidth*. Penulis menggunakan algoritma *Lempel Ziv Welch* dalam proses kompresi dan dekompresi, dimana masukannya adalah *file* citra. Penulis menerapkan metode *prototype*, metode yang biasa digunakan dalam pengembangan perangkat lunak. Penelitian ini menggunakan 15 uji coba kompresi, hasil yang didapat dari rasio rata-rata citra BMP adalah 60,95%. Kemudian untuk rasio rata-rata citra SVG adalah 111,36%. Untuk rasio rata-rata citra TIFF adalah 71,93%. Pada saat yang sama, kecepatan proses kompresi dan dekompresi sebanding dengan ukuran *file*, artinya semakin besar *file* yang diproses, semakin lama prosesnya. Untuk hasil uji coba dekompresi, *file* kembali ke ukuran semula dan dapat dikatakan proses dekompresi berhasil.

Kata kunci : Kompresi, Dekompresi, Algoritma, Perangkat Lunak, Lempel Ziv Welch

1. PENDAHULUAN

Citra digital kini banyak digunakan di berbagai bidang. Mulai dari kebutuhan sehari-hari seperti percetakan foto, konten *website*, konten media sosial, pemetaan hutan, identifikasi forensik, rekam medis menggunakan citra medis, dan citra satelit. Hampir semua citra digital membutuhkan media penyimpanan yang cukup besar [1]. Oleh karena itu, timbul masalah jika citra disimpan dalam *database* dengan media penyimpanan yang terbatas. Permasalahan lain adalah jika ingin menggunakan jalur komunikasi atau internet untuk mengirimkan citra digital. Citra digital juga membutuhkan waktu pengiriman yang lebih lama karena ukurannya yang besar. Oleh karena itu, dicari teknik yang dapat memperkecil ukuran *file* citra digital melalui kompresi. Hampir semua teknik kompresi memiliki keuntungan dalam mengurangi ukuran *file* gambar digital, sehingga meminimalkan semua batasan di atas [2].

Selain itu dengan adanya media penyimpanan berkapasitas besar berarti orang tidak lagi bermasalah jika memiliki *file* berukuran besar. Juga jika *file* yang mereka miliki adalah *file* gambar. Namun terkadang *file* berukuran besar merepotkan jika harus mengelola *file* media penyimpanannya sendiri untuk berbagai jenis data [1]. Selain itu, jika *file* tersebut dikirim secara elektronik, tentu kapasitas *file* itu sendiri menjadi masalah [1]. Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan *bandwidth* [3]. Hal yang sama berlaku untuk Rumah Produksi Delvia.

Rumah Produksi Delvia adalah sebuah instansi yang bergerak di bidang konveksi dan digital *printing*.

Salah satu jasa Rumah Produksi Delvia adalah berfokus pada bidang percetakan digital *printing*. Dalam melakukan percetakan dibutuhkan sebuah desain, desain ini bisa dari klien ataupun klien yang memesan desain ke admin Delvia. Dalam pembuatan desain biasanya dibutuhkan konten berupa gambar-gambar yang kualitas tinggi untuk menjadikannya lebih menarik dan meningkatkan pengalaman klien. Setelah desain selesai dibuat akan dilakukan *export* desain, desain biasanya di *export* dalam bentuk *file* citra SVG, BMP, dan TIFF. *File* citra hasil *export* dari desain tersebut memiliki ukuran yang cukup besar, yaitu lebih dari 10 *Megabyte* (MB).

Hal tersebut menyebabkan kendala dalam penyimpanan *file* untuk dijadikan arsip, karena *file* berukuran besar merepotkan jika harus mengelola *file* media penyimpanannya. Kendala lain yaitu terjadinya peningkatan penggunaan data yang telah menyebabkan masalah pada penyimpanan data dan secara tidak langsung meningkatkan permintaan akan penyimpanan data. Semakin besar ukuran *file*, semakin banyak ruang penyimpanan yang dibutuhkan.

Terkait dengan masalah besarnya ukuran *file*, salah satu solusi yang telah dikenal untuk masalah ukuran *file* besar adalah teknologi kompresi. Teknik ini mengurangi ukuran *file* besar sebelum dikirim tanpa mengurangi nilai informasi yang terkandung di dalamnya. Hasilnya waktu transfer data menjadi lebih cepat.

Kompresi citra adalah proses meminimalkan jumlah bit yang mewakili gambar untuk membuatnya lebih kecil ukurannya [4]. Pada dasarnya, teknik kompresi citra digunakan dalam proses transmisi data dan penyimpanan data [5]. Kompresi citra banyak

digunakan dalam penyiaran televisi, penginderaan jauh, komunikasi militer, radar dan bidang lainnya [5]. Tujuan dari kompresi citra sendiri untuk mengurangi redundansi dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien [4].

Saat ini telah dikembangkan berbagai macam algoritma kompresi data, seperti algoritma kompresi *lossless* di antaranya adalah *Huffman*, *delta*, *run-length encoding*, dan *Lempel-Ziv-Welch* (LZW) [1]. Salah satu algoritma yang sering digunakan adalah algoritma *Lempel-Ziv-Welch* (LZW), algoritma *Lempel-Ziv-Welch* (LZW) banyak digunakan untuk mengkompresi data digital, seperti *file* citra, dan *file* teks [6]. Algoritma *Lempel-Ziv-Welch* (LZW) ini bersifat *lossless*, sehingga cocok untuk kompresi citra yang mengandung informasi penting yang tidak boleh rusak akibat kompresi [7]. Pada penelitian ini algoritma *Lempel-Ziv-Welch* (LZW) akan dikembangkan untuk mengkompres *file* yang bertipe citra, yaitu SVG, BMP, dan TIFF. Algoritma *Lempel-Ziv-Welch* (LZW) ini melakukan kompresi dengan menggunakan *dictionary*, dimana fragmen-fragmen teks diganti dengan indeks yang diperoleh oleh sebuah kamus [8]. Kelebihan dari algoritma *Lempel Ziv Welch* (LZW) yaitu cepat dalam implementasi dan kekurangannya kurang optimal karena hanya melakukan analisis terbatas pada data [7].

Pada perancangan aplikasi, penulis menggunakan algoritma *Lempel-Ziv-Welch* (LZW) untuk algoritma dari aplikasi kompresi citra yang akan dibuat. Penggunaan algoritma *Lempel-Ziv-Welch* (LZW) digunakan karena algoritma ini bersifat *lossless*, sehingga ukuran citra yang dikompresi dapat diperkecil tanpa kehilangan kualitas citra [7].

Beberapa penelitian sebelumnya, bahwa Hasil penilaian kompresi menggunakan algoritma *Lempel Ziv Welch* (LZW) menunjukkan tingkat rata-rata rasio kompresi dan untuk semua jenis *file* teks sebesar 51,04% dengan rata-rata 2,56 detik, untuk jenis *file* gambar sebesar 37,26% dengan rata-rata waktu 0,44 detik [9]. Kemudian teknik kompresi *Lempel Ziv Welch* (LZW) dapat memberikan penghematan penyimpanan setiap artikel berkisar antara 11,6656% sampai dengan 16,6656% untuk perhitungan nilai rata-rata persentase penghematan 200 sampai dengan 1.000 karakter pertama dari sepuluh buah artikel acak [10].

Perancangan ini akan dilakukan menggunakan algoritma *Lempel Ziv Welch* (LZW). Aplikasi kompresi citra ini diharapkan dapat membantu dalam mengelola penyimpanan *file* serta mempersingkat waktu pengiriman *file* ke klien untuk memeriksa *file* desain sebelum dicetak. Hasil aplikasi ini adalah *website* kompresi citra yang mendukung format *file* citra SVG, BMP, dan TIFF.

2. TINJAUAN PUSTAKA

Sebelum penelitian ini, terdapat penelitian-penelitian terdahulu tentang kompresi, kompresi citra, dan penelitian terkait tentang algoritma *Lempel-Ziv-*

Welch (LZW) pada aplikasi kompresi citra dengan menggunakan algoritma *Lempel-Ziv-Welch* (LZW). Uraian dari beberapa penelitian sebelumnya yang dilakukan oleh peneliti adalah sebagai berikut.

Analisis Perbandingan Algoritma LZW dan *Huffman* Pada Kompresi *File* Gambar Bmp dan Png. Penelitian ini membahas tentang analisis perbandingan antara algoritma LZW dan *Huffman* terhadap *file* citra terkompresi berformat BMP dan PNG. Penelitian ini bertujuan untuk memberikan pengetahuan tentang algoritma apa yang terbaik antara kedua algoritma tersebut untuk kompres *file* gambar format BMP dan PNG. Penelitian ini membandingkan rasio dan estimasi waktu dari masing-masing algoritma untuk *file* gambar berformat BMP dan PNG. Berdasarkan pengujian dan analisis percobaan proses kompresi, maka diperoleh kesimpulan bahwa pada proses kompresi *file* berformat BMP dan PNG, algoritma *Huffman* dapat mengkompres *file* lebih cepat dibandingkan algoritma LZW, namun algoritma LZW dapat menghasilkan rasio kompresi yang lebih baik dibandingkan algoritma *Huffman* [7].

Implementasi Algoritma Kompresi LZW Pada *Database Server*. Penelitian ini membahas tentang implementasi algoritma kompresi LZW untuk mengkompresi data yang tersimpan dalam web server. Hal tersebut dilakukan untuk memperkecil ukuran data yang akan disimpan di web server. Proses kompresi dilakukan ketika data (artikel) disimpan di server web dan didekompresi setiap kali artikel dibaca. Hasil implementasi algoritma LZW pada web server untuk penyimpanan artikel semakin meningkat, hal ini dapat dilihat dari nilai persentase penyimpanan setiap artikel berkisar antara 11,6656% sampai dengan 16,6656% untuk perhitungan nilai rata-rata penghematan persentase 200 hingga 1000 karakter pertama dari sepuluh buah artikel acak [10].

Kompresi Citra Dengan Menggunakan Metode *Delta Modulation*. Kompresi citra adalah teknik yang digunakan untuk mengurangi biaya penyimpanan dan transmisi. Kompresi citra merupakan metode yang sangat berguna untuk pengembangan citra digital. Dengan kompresi, data citra digital yang berukuran besar dapat dikompresi sehingga memiliki ukuran yang lebih kecil. Program kompresi citra ini menggunakan metode *Delta Modulation*, dibuat dengan GUI dan *source coding*. Dengan menerapkan fungsi *Dmsig* untuk pemrosesan sinyal, diperoleh rumus untuk mengompresi citra yang dipengaruhi oleh ukuran langkah dan ukuran citra. Rata-rata hasil kompresi citra yang diperoleh dari setiap format citra memiliki hasil yang berbeda-beda. Gambar dengan format JPG memiliki rasio kompresi paling tinggi diantara format-format gambar lain [5].

Penerapan Algoritma LZW dan Kuantisasi Dalam Kompresi Citra Digital. Pada penelitian ini membahas tentang penerapan algoritma LZW dan kuantisasi dalam program kompresi citra digital. Tujuan dari penelitian ini untuk mengatasi ukuran suatu *file* yang besar sehingga masalah penyimpanan,

pengiriman data maupun kebutuhan *bandwidth* yang digunakan dapat diatasi. Pada penelitian ini diperoleh hasil, bahwa citra kompresi algoritma kuantisasi dan algoritma *Lempel Zip Welch* (LZW) baik untuk kompresi *file* citra. Kemudian *file* citra yang telah dikompresi dengan menggunakan algoritma kuantisasi tidak dapat dikembalikan lagi ke data awal. Algoritma *Lempel Zip Welch* (LZW) kurang baik diterapkan pada *file* citra. Algoritma kuantisasi jauh lebih baik dari pada algoritma *Lempel Ziv Welch* untuk *file* citra [8].

Optimasi Web Service dengan Penerapan Algoritma Kompresi LZW: Studi Kasus Aplikasi BluCampus Universitas Budi Luhur. Pada penelitian ini membahas penerapan web service berbasis REST sebagai media pertukaran data dan algoritma kompresi LZW untuk mengurangi ruang penyimpanan pada database server. Pada penelitian ini akan dianalisis semua web service yang akan digunakan pada aplikasi android beserta kelebihan dan kekurangan algoritma kompresi pada data teks yang berjumlah 100 sampai dengan 1600 karakter. Hasil penelitian menunjukkan bahwa semua 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* [11].

Optimasi Rasio Kompresi Dan Kompleksitas Waktu Kompresi File Teks Menggunakan Algoritma *Lempel-ZIV-Welch* Dengan *Fibonacci Search*. Pada penelitian ini, Algoritma *Fibonacci Search* (FS) diterapkan untuk meningkatkan kinerja algoritma LZW dalam hal pencarian kamus kata (*dictionary*), sehingga dengan pencarian yang optimal akan diperoleh kinerja LZW yang lebih baik. Data yang digunakan dalam penelitian ini adalah data teks dengan alasan data teks lebih sederhana dalam pengolahannya. Hasil penelitian ini menguji lima jenis data teks menggunakan algoritma LZW, dengan kapasitas yang berbeda yaitu 6, 9, 12, 19 dan 24 Kb diperoleh rata – rata ukuran *file* kompresi sebesar 4,148, rata-rata waktu kompresi sebesar 118,8, rata-rata rasio kompresi 64%. Dengan metode LZWFS diperoleh rata-rata ukuran *file* kompresi 5,126, rata-rata waktu kompresi 86,2, rata – rata rasio kompresi 58%. Dari hasil penelitian di atas diperoleh kesimpulan bahwa algoritma LZWFS berhasil menyingkat waktu pencarian data namun masih memiliki kelemahan pada saat pengurutan data [6].

Kompresi File Menggunakan Algoritma *Lempel Ziv Welch* (LZW). Dalam bidang teknologi informasi, komunikasi data erat kaitannya dengan transmisi *file*. Terkadang ukuran *file* menjadi kendala dalam melakukan proses pengiriman. *File* besar akan memakan waktu lebih lama untuk dikirim daripada *file* yang lebih kecil. Oleh karena itu, untuk mengatasi masalah tersebut salah satunya adalah kompresi. Aplikasi ini menerapkan algoritma *Lempel Ziv Welch*

(LZW). Algoritma *Lempel Ziv Welch* (LZW) termasuk dalam teknik kompresi *lossless*, yang merupakan teknik kompresi yang tidak mengubah data asli. Hasil penilaian kompresi menggunakan algoritma *Lempel Ziv Welch* (LZW) menunjukkan tingkat rata-rata rasio kompresi dan untuk semua jenis *file* teks sebesar 51,04% dengan rata-rata 2,56 detik, untuk jenis *file* gambar sebesar 37,26% dengan rata-rata waktu 0,44 detik. Berdasarkan persentase rata-rata rasio kompresi untuk semua jenis *file* yang diuji menggunakan algoritma *Lempel Ziv Welch* (LZW) adalah 40,40% dengan waktu rata-rata yang diperlukan adalah 1,81 detik [9].

Implementasi Algoritma *Run Length Encoding* Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra. Ukuran data yang besar terkadang menjadi kendala dalam proses pengiriman data. Data dengan ukuran besar akan membutuhkan waktu transfer yang lebih lama dibandingkan data dengan ukuran yang lebih kecil, terkadang terdapat risiko tidak dapat tertampung pada media penyimpanan dan tidak terkirim, sehingga akan mengurangi kapasitas kosong pada media penyimpanan. Tujuan dari penelitian ini bertujuan untuk meminimalkan kebutuhan memori dalam mempresentasikan citra digital dengan mengurangi duplikasi data. Hasil dari teknik kompresi *Run Length Encoding* dapat menghasilkan *ratio* kompresi sebesar 11,47% dari *file* citra yang awalnya 103.38 Kb menjadi 91.53 Kb [12].

Art Image Compression Based on Lossless LZW Hashing Ciphering Algorithm. Kompresi gambar berwarna adalah cara yang baik untuk mengkodekan gambar digital dengan mengurangi jumlah *bit* yang diinginkan untuk memasok gambar. Tujuan untuk mengurangi ruang penyimpanan, mengurangi biaya transportasi dan menjaga kualitas yang baik. Dalam penelitian ini, metodologi sederhana yang efektif diusulkan untuk tujuan mengompresi gambar digital seni warna dan memperoleh *bit rate* yang rendah dengan mengompresi matriks yang dihasilkan dari proses kuantisasi skalar (mengurangi jumlah *bit* dari 24 menjadi 8 *bit*) menggunakan perpindahan coding dan kemudian mengompresi sisanya menggunakan algoritma *Mabel ZF Welch* LZW. Metodologi yang diusulkan mempertahankan kualitas gambar yang direkonstruksi. Hasil penelitian dari makroskopik dan kuantitatif pada citra teknis berwarna menunjukkan bahwa metodologi yang diusulkan memberikan citra hasil rekonstruksi dengan nilai PSNR yang tinggi dibandingkan dengan teknik kompresi citra standar [13].

An Efficient Lossless Medical Data Compression using LZW compression for Optimal Cloud Data Storage. Di era digital saat ini, rekam medis pasien semua diproses dan disimpan secara digital. Dengan meningkatnya populasi, kebutuhan ruang penyimpanan untuk data medis meningkat secara eksponensial. Perusahaan medis dan farmasi menghabiskan banyak uang di ruang penyimpanan

untuk menyimpan catatan pasien. Penelitian ini menyajikan teknik kompresi data *lossless* baru yang cepat dan efisien untuk menyimpan data medis. Kompresi LZW adalah teknik kompresi *lossless* berbasis kamus yang memanfaatkan redundansi dan pengulangan dalam data untuk mengompresnya. Metode yang diusulkan dibandingkan dengan teknik kompresi *lossless* lainnya. Metode yang diusulkan diimplementasikan menggunakan python dan HADOOP digunakan untuk menyimpan data terkompresi. Kompresi LZW yang digunakan dalam metode yang diusulkan mengompresi data medis lebih baik daripada *run length coding* dan *huffman coding*. Hasil eksperimen menunjukkan bahwa metode kompresi yang diusulkan mengungguli metode kompresi yang ada [14].

Dari masing-masing penelitian di atas, terdapat kesamaan dengan penelitian yang dilakukan oleh peneliti yaitu melakukan kompresi data, dengan menggunakan algoritma *Lempel Ziv Welch* (LZW), namun dengan subjek, objek, dan lokasi penelitian yang berbeda. Penelitian sebelumnya yang memiliki banyak kesamaan dengan penelitian ini adalah penelitian yang dilakukan oleh Jorellis Bakara pada 2017 yaitu penelitian tentang Implementasi Algoritma LZW dan Kuantisasi Dalam Kompresi Citra Digital. Diantara dua penelitian ini terdapat beberapa kesamaan yaitu penggunaan algoritma *Lempel Ziv Welch* (LZW) sebagai teknik kompresi dan citra sebagai fokus utama aplikasi. Namun penelitian yang diajukan pada proposal ini memiliki beberapa kelebihan dibanding penelitian yang dilakukan oleh Jorellis Bakara yaitu aplikasi pada penelitian ini berbasis web, kemudian bisa digunakan di perangkat yang berbeda seperti *computer*, *laptop*, *smartphone*, dan *computer tablet*, dan pengguna lain bisa mengakses aplikasi.

3. METODE PENELITIAN

Metode penelitian digunakan sebagai pedoman untuk menentukan langkah-langkah, uraian, prosedur, waktu dan tempat pengumpulan data. Metode yang akan digunakan dalam penelitian ini adalah metode *Research and Development* (R&D), kemudian untuk model yang digunakan adalah *prototype*, karena hasil akhir penelitian ini akan menghasilkan produk aplikasi kompresi citra.

Namun, tahapan *prototype* yang akan dilakukan pada penelitian ini hanya akan berhenti pada tahap ke-5 yaitu *refining prototype* (memperbaiki prototipe). Berikut akan dijelaskan langkah-langkah penelitian yang akan dilakukan sesuai dengan tahapan model *prototype*.

- a. *Requirements Gathering and Analysis* (Analisis Kebutuhan)

Untuk mengembangkan suatu sistem, diperlukan penilaian kebutuhan awal dan analisis suatu ide atau gagasan untuk mengembangkan sistem tersebut. Analisis dilakukan untuk mengetahui komponen apa saja yang berjalan dalam sistem. Pada tahap ini peneliti mengidentifikasi kebutuhan sistem.

- b. *Quick Design* (Desain cepat)

Pada tahap ini memberikan gambaran singkat dari sistem yang ingin dibuat. Tentunya berdasarkan diskusi dari tahap pertama di awal. Dalam tahap desain cepat ini akan diawali dengan membuat *flowchart*, struktur tabel *database*, dan desain *layout*.

- c. *Build Prototype* (Membangun Prototipe)

Selama fase *prototyping*, fokusnya adalah menunjukkan kepada pengguna bagaimana sistem akan mengalir dan terlihat, dan pada tahap ini hanya desain sementara. Dalam tahap membangun *prototype* ini akan diawali dengan membuat diagram konteks, *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD).

- d. *User Evaluation* (Evaluasi Pengguna Awal)

Pada tahap ini peneliti akan memvalidasi aplikasi yang telah dibangun kepada validator. Proses evaluasi ini dilakukan oleh pengguna untuk mengetahui apakah prototipe yang dibangun sesuai dengan keinginan pengguna. Dari hasil pengujian yang dilakukan, apakah harapan pengguna terpenuhi, jika sesuai sistem dapat digunakan, jika tidak, ulangi langkah dua dan tiga.

- e. *Refining Prototype* (Memperbaiki Prototipe)

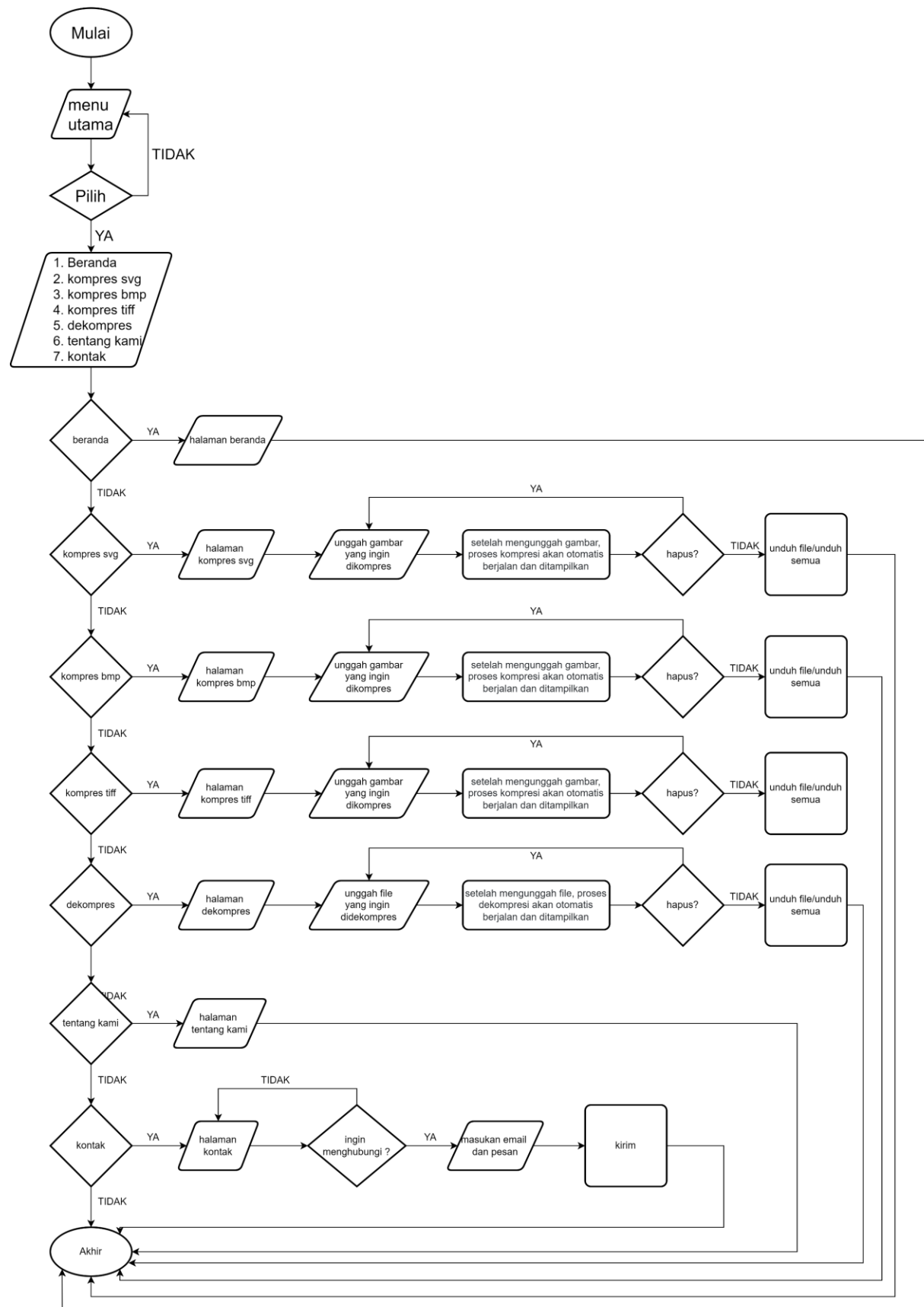
Pada tahap ini peneliti akan memperbaiki aplikasi dengan mempertimbangkan saran dan masukan dari validator. Namun, jika pengguna memiliki catatan perbaikan sistem, maka tahap 4-5 akan terus berulang hingga pengguna menyetujui sistem yang akan dikembangkan.

- f. *Implement Product and Maintain* (Implementasi dan Pemeliharaan)

Pada tahap ini sebenarnya *product* itu sudah siap untuk digunakan atau diujicobakan. Namun, pada penelitian ini proses implementasi *product* dan *maintain* tidak dilakukan, karena aplikasi belum diterapkan secara nyata kepada *user*, serta masih dibutuhkan pengujian lebih lanjut.

3.1. Flowchart Sistem

Pada gambar 1 merupakan *flowchart* sistem. Berdasarkan hasil analisa, ini merupakan *flowchart* sistem untuk pengguna aplikasi kompresi citra, berikut ini adalah proses alur dari tampilan web pengguna ataupun pengunjung web. Untuk halaman yang disajikan terdapat tujuh menu yaitu beranda, kompres *svg*, kompres *bmp*, kompres *tiff*, dekompresi, tentang kami, dan kontak.



Gambar 1. Flowchart sistem

3.2. Struktur Tabel Database

Tabel ini digunakan untuk menyimpan *history* pengguna. Pada tabel ini diberikan kolom *created at* dengan tipe data *timestamp*, sehingga *history* pengguna yang menggunakan tercatat di *database* dapat dilihat pada Tabel 1 berikut:

Tabel 1. Struktur Tabel Pengguna

Field	Type	Null	Key
id	bigint(20)	no	PK
ip_address	varchar(20)	null	
statistic_date	date	null	
hits	int(11)	0	
online	varchar(150)	null	
device	varchar(150)	null	
browser	text	null	
platform	text	null	
user_agents	text	null	
created_at	timestamp	null	
updated_at	timestamp	null	

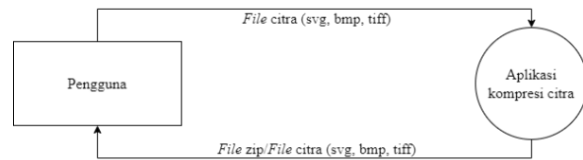
Tabel ini digunakan untuk menyimpan data citra. Pada tabel ini diberikan kolom nama yang digunakan untuk menyimpan nama *file* ketika diunggah. Kemudian terdapat kolom *path* yang digunakan untuk menyimpan jalur dari lokasi penyimpanan *file*. Kolom *created* dan *updated* digunakan untuk menyimpan informasi kapan *file* tersebut dibuat dan kapan terakhir kali *file* tersebut diperbaharui., dapat dilihat pada Tabel 2 berikut:

Tabel 2. Struktur Tabel Unggah Citra

Field	Type	Null	Key
id	int(10)	No	PK
img_filename	varchar(255)	No	
img_original_file_name	varchar(255)	Yes	
img_path	varchar(255)	Yes	
img_extension	varchar(10)	Yes	
img_mime	varchar(255)	Yes	
img_size	decimal(10,2)	Yes	
img_execution_compress_time	decimal(10,5)	Yes	
img_execution_decompress_time	decimal(10,5)	Yes	
img_compress_size	decimal(10,2)	Yes	
img_compress_ratio	decimal(10,2)	Yes	
created_at	timestamp	Yes	
updated_at	timestamp	Yes	

3.3. Diagram Konteks

Untuk menggambarkan bagaimana hubungan antara entitas luar, *input* dan *output* sistem, maka dibuat diagram konteks. Diagram konteks aplikasi kompresi citra menggunakan algoritma *Lempel-Ziv-Welch* yang akan dirancang dapat dilihat pada Gambar 2.

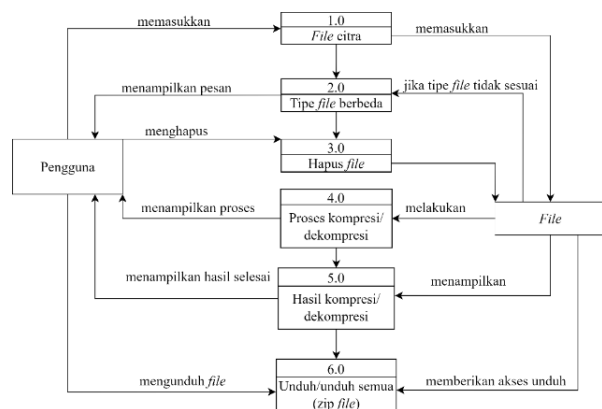


Gambar 2. Diagram Konteks

Pada diagram konteks Gambar 2 ini, pengguna melakukan unggah citra berdasarkan ekstensi yang diizinkan pada sistem. Kemudian proses unggah citra berjalan dan dilanjutkan dengan proses kompresi. Setelah itu akan muncul tombol unduh, pengguna bisa mengunduh *file* secara langsung atau mengunduh *file* secara keseluruhan dalam bentuk format zip.

3.4. Data Flow Diagram (DFD)

Berikut ini merupakan proses DFD level 1, yaitu setiap tahap yang ada dalam suatu proses yang dapat digambarkan menjadi lebih detail. DFD level 1 dari aplikasi kompresi citra dengan menggunakan algoritma *Lempel-Ziv-Welch* dapat dilihat pada Gambar 3.

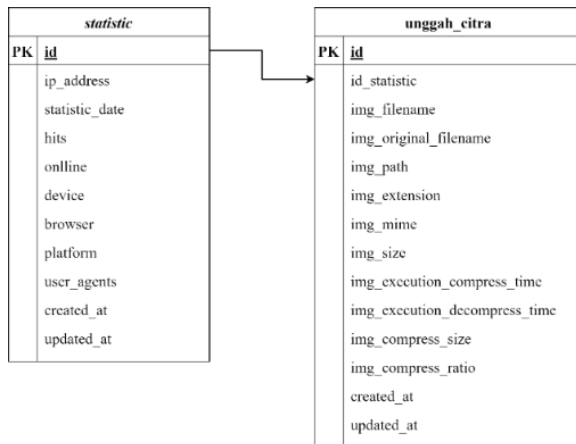


Gambar 3. Data Flow Diagram Level 1

Proses tersebut terdiri dari enam tahap, yaitu memasukkan *file* citra, menampilkan pesan jika *file* citra tidak sesuai, menghapus *file* citra, jika *file* citra sesuai maka akan dilanjutkan ke proses kompresi, menampilkan hasil kompresi, dan mengunduh *file*.

3.5. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) aplikasi kompresi citra dengan menggunakan algoritma *Lempel-Ziv-Welch*. Dalam melakukan kompresi citra dapat dilakukan menggunakan sistem yang berupa aplikasi program untuk melakukan pemampatan pada *size file* citra. Beberapa entitas yang ada di aplikasi program atau terhubung menjadi suatu susunan yang membentuk struktur *database* aplikasi kompresi citra dengan menggunakan algoritma *Lempel-Ziv-Welch* dapat dilihat pada Gambar 4.

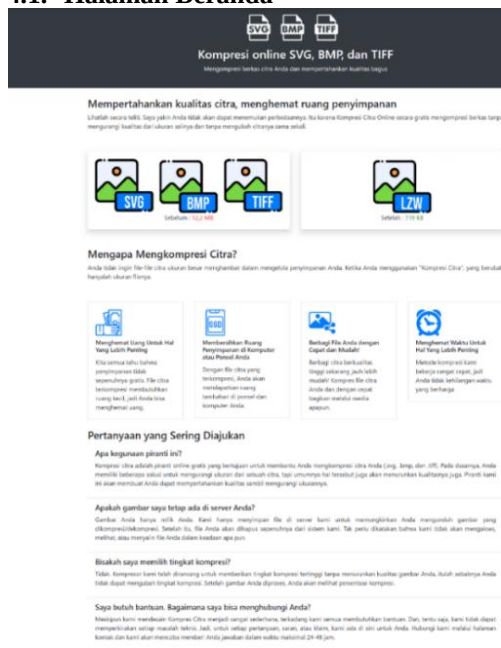


Gambar 4. Entity Relationship Diagram

4. HASIL DAN PEMBAHASAN

Pada tahap ini hasil rancangan akan diimplementasikan menjadi sebuah aplikasi berbasis web yang dibangun menggunakan *framework* Laravel, sistem ini memiliki dua proses yaitu proses kompresi dan dekompresi.

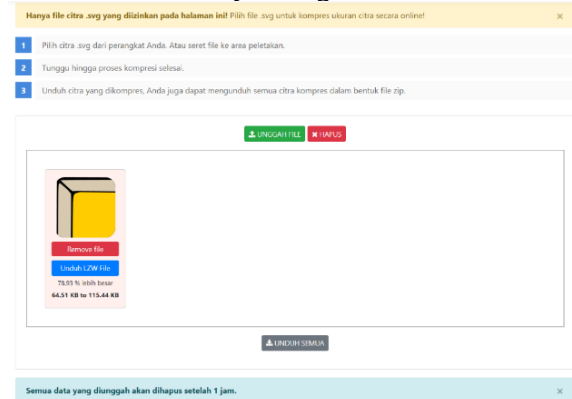
4.1. Halaman Beranda



Gambar 5. Konten Halaman Beranda

Pada halaman beranda bagian konten terdapat beberapa macam bagian, bagian konten terdiri dari tipe *file* citra yang dapat dikompresi, contoh hasil kompres citra sebelum dan sesudah, mengapa harus mengompresi citra?, dan pertanyaan-pertanyaan yang diajukan oleh pengguna, seperti yang ada pada Gambar 5.

4.2. Halaman Kompres Svg

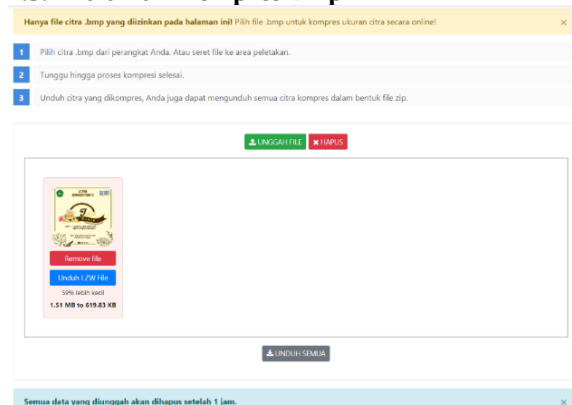


Gambar 6. Halaman Kompres Svg

Halaman kompres svg ini merupakan halaman yang digunakan untuk kompresi *file* citra dengan ekstensi .SVG. Pada halaman ini diberi tampilan *alert* agar pengguna mengerti, bahwa pada halaman ini hanya diizinkan untuk kompres *file* citra dengan ekstensi .SVG saja. Kemudian pada halaman ini terdapat *list* langkah-langkah dalam melakukan kompresi *file* citra, *list* ini dapat membantu pengguna agar mengerti dalam melakukan kompresi *file* citra. Kemudian terdapat *form* unggah citra, pada *form* ini terdapat tombol unggah citra, tombol hapus dan tombol unduh semua. Selain itu terdapat juga tampilan *alert* yang memberi tahu bahwa data yang diunggah akan dihapus setelah satu jam.

Pengguna dapat melakukan kompresi *file* dengan menekan tombol unggah atau melakukan *drag and drop* *file* citra ke bagian *form* kompresi. Setelah itu proses unggah *file* berjalan, dan dilanjutkan dengan proses kompresi. Setelah proses unggah dan kompresi selesai, maka *file* dapat diunduh, selain itu pada bagian bawah tombol unduh *file* terdapat rasio citra dan *size* citra yang telah dikompresi. *File* keluaran terkompresi yang dilakukan menggunakan algoritma *Lempel-Ziv-Welch* (LZW) diberi ekstensi .LZW tambahan, seperti pada Gambar 6.

4.3. Halaman Kompres Bmp

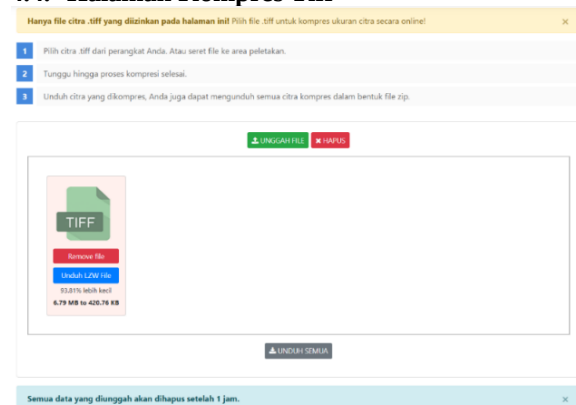


Gambar 7. Halaman Kompres Bmp

Pada halaman kompres bmp ini merupakan halaman yang digunakan untuk kompresi *file* citra dengan ekstensi .BMP. Pada halaman ini diberi tampilan *alert* agar pengguna mengerti, bahwa pada halaman ini hanya diizinkan untuk kompres *file* citra dengan ekstensi .BMP saja. Kemudian pada halaman ini terdapat *list* langkah-langkah dalam melakukan kompresi *file* citra, *list* ini dapat membantu pengguna agar mengerti dalam melakukan kompresi *file* citra. Kemudian terdapat *form* unggah citra, pada *form* ini terdapat tombol unggah citra, tombol hapus dan tombol unduh semua. Selain itu terdapat juga tampilan *alert* yang memberi tahu bahwa data yang diunggah akan dihapus setelah satu jam.

Pengguna dapat melakukan kompresi *file* dengan menekan tombol unggah atau melakukan *drag and drop file* citra ke bagian *form* kompresi. Setelah itu proses unggah *file* berjalan, dan dilanjutkan dengan proses kompresi. Setelah proses unggah dan kompresi selesai, maka *file* dapat diunduh, selain itu pada bagian bawah tombol unduh *file* terdapat rasio citra dan *size* citra yang telah dikompresi. *File* keluaran terkompresi yang dilakukan menggunakan algoritma *Lempel-Ziv-Welch* (LZW) diberi ekstensi .LZW tambahan, seperti pada Gambar 7.

4.4. Halaman Kompres Tiff



Gambar 8. Halaman Kompres Tiff

Pada halaman kompres tiff ini merupakan halaman yang digunakan untuk kompresi *file* citra dengan ekstensi .TIFF. Pada halaman ini diberi tampilan *alert* agar pengguna mengerti, bahwa pada halaman ini hanya diizinkan untuk kompres *file* citra dengan ekstensi .TIFF saja. Kemudian pada halaman ini terdapat *list* langkah- langkah dalam melakukan kompresi *file* citra, *list* ini dapat membantu pengguna agar mengerti dalam melakukan kompresi *file* citra. Kemudian terdapat *form* unggah citra, pada *form* ini terdapat tombol unggah citra, tombol hapus dan tombol unduh semua. Selain itu terdapat juga tampilan *alert* yang memberi tahu bahwa data yang diunggah akan dihapus setelah satu jam.

Pengguna dapat melakukan kompresi *file* dengan menekan tombol unggah atau melakukan *drag and drop file* citra ke bagian *form* kompresi. Setelah itu proses unggah *file* berjalan, dan dilanjutkan dengan proses kompresi. Setelah proses unggah dan kompresi selesai, maka *file* dapat diunduh, selain itu pada bagian bawah tombol unduh *file* terdapat rasio citra dan *size* citra yang telah dikompresi. *File* keluaran terkompresi yang dilakukan menggunakan algoritma *Lempel-Ziv-Welch* (LZW) diberi ekstensi .LZW tambahan, seperti pada Gambar 8.

4.5. Halaman Dekompresi

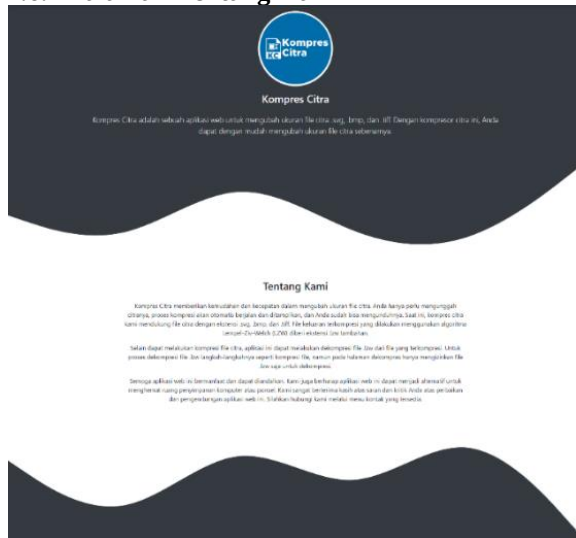


Gambar 9. Halaman Dekompres

Halaman dekompres ini merupakan halaman yang digunakan untuk dekompresi *file* ekstensi .LZW, yang mana untuk mengembalikan *file* citra yang telah dikompres menjadi seperti semula tanpa ekstensi .LZW. *File* dengan ekstensi .LZW ketika didekompresi akan menjadi *file* citra yang sebelumnya, jika *file* .LZW awalnya citra .BMP, maka setelah didekompresi akan menjadi .BMP lagi, begitu juga dengan *file* .SVG dan .TIFF. Pada halaman ini diberi tampilan *alert* agar pengguna mengerti, bahwa pada halaman ini hanya diizinkan untuk kompres *file* dengan ekstensi .LZW saja. Kemudian pada halaman ini terdapat *list* langkah-langkah dalam melakukan kompresi *file* .LZW, *list* ini dapat membantu pengguna agar mengerti dalam melakukan kompresi *file* .LZW. Kemudian terdapat *form* unggah *file*, pada *form* ini terdapat tombol unggah citra, tombol hapus dan tombol unduh semua. Selain itu terdapat juga tampilan *alert* yang memberi tahu bahwa data yang diunggah akan dihapus setelah satu jam.

Pengguna dapat melakukan dekompresi *file* dengan menekan tombol unggah atau melakukan *drag and drop file* ke bagian *form* dekompresi. Setelah itu proses unggah *file* berjalan, dan dilanjutkan dengan proses dekompresi. Setelah proses unggah dan dekompresi selesai, maka *file* dapat diunduh. *File* keluaran dekompresi yang dilakukan menggunakan algoritma *Lempel-Ziv-Welch* (LZW) dikembalikan sesuai ekstensi dari citra, seperti pada Gambar 9.

4.6. Halaman Tentang Kami



Gambar 10. Halaman Tentang Kami

Halaman tentang kami merupakan halaman yang berisi informasi, dari nama aplikasi web, deskripsi aplikasi web, dan cara penggunaan. Jadi halaman ini merupakan informasi untuk pengunjung agar mengetahui atau lebih tahu tentang aplikasi web, seperti pada Gambar 10.

4.7. Halaman Kontak

Gambar 11. Halaman Kontak

Halaman kontak ini merupakan halaman yang digunakan oleh pengguna untuk mengirimkan pesan ke pengembang aplikasi. Jadi halaman ini adalah sebuah sarana bagi pengunjung untuk menghubungi pengembang aplikasi, baik untuk menanyakan perihal yang penting atau memberikan kritik dan saran secara pribadi, seperti yang ada pada Gambar 11.

4.8. Pengujian Validator

Uji validator dilakukan untuk menguji apakah aplikasi sesuai dengan metode yang dipilih. Penguji terdiri dari ahli IT dan ahli Desain Grafis. Tim penguji untuk uji validator, antara lain:

Tabel 3. Pihak Penguji Vlidator

No.	Nama Penguji	Sebagai	Profesi
1.	Rifky Ramadhan	Ahli IT	Junior Software Engineer
2.	M. Anto	Ahli IT	IT Developer
3.	Rian Agi Septian Putra	Ahli Desain Grafis	Desainer Grafis

Setelah menentukan penguji validator sebagai ahli IT dan Desain Grafis, maka dilanjutkan dengan menyiapkan pernyataan terkait pengujian validator. Berikut ini merupakan tabel pernyataan untuk pengujian validator:

Tabel 4. Pengujian Validasi Ahli

Aspek	kode	Pernyataan
Learnability	A1	Aplikasi kompresi citra dapat mudah dipelajari
	A2	Saya dapat dengan mudah menerima informasi dari aplikasi
	A3	Saya dapat dengan mudah memahami dan mengerti alur aplikasi
	A4	Tanpa informasi tertulis, saya dapat mengerti penggunaan aplikasi kompresi citra
Memorability	B1	Saya dapat dengan mudah mengingat penggunaan aplikasi kompresi citra ini
	B2	Saya dapat dengan mudah mengingat navigasi pada aplikasi kompresi citra
Efficiency	C1	Saya mampu mengakses menu pada aplikasi dengan cepat
	C2	Saya dapat dengan mudah memperoleh informasi yang saya butuh kan pada aplikasi kompresi citra
	C3	Saya dapat menemukan fitur dan informasi yang saya cari ketika awal membuka aplikasi kompresi citra
	C4	Aplikasi dapat melakukan kompresi citra dengan format SVG dan ukuran file berhasil terkompresi
	C5	Aplikasi dapat melakukan kompresi citra dengan format BMP dan ukuran file berhasil terkompresi
	C6	Aplikasi dapat melakukan kompresi citra dengan format TIFF dan ukuran file berhasil terkompresi
	C7	Aplikasi dapat melakukan dekomposisi file LZW dan file berhasil terdekompresi
Error	D1	Saya tidak menemukan <i>error</i> ketika menggunakan aplikasi kompresi citra ini
	D2	Saya tidak menemukan <i>error</i> pada fungsi dan menu nya
	D3	Saya tidak menemukan <i>error</i> ketika melakukan kompresi file
	D4	Saya tidak menemukan <i>error</i> ketika melakukan dekomposisi file
Satisfaction	E1	Saya senang dengan desain antar muka aplikasi kompresi citra
	E2	Saya merasa nyaman menggunakan aplikasi kompresi citra
	E3	Paduan tata letak dan warna nyaman untuk dilihat
	E4	Aplikasi kompresi citra sesuai dengan ekspektasi saya ketika melihat judulnya

Berdasarkan hasil uji validasi ahli IT dan ahli dibidang Desain Grafis, hasil yang diperoleh dengan nilai rata-rata 74%, menunjukkan bahwa aplikasi kompresi layak. Namun demikian, aplikasi kompresi citra masih harus dikembangkan dari segi informasi tertulis, navigasi, informasi aplikasi, kompresi *file* citra SVG, dekompresi *file*, *error* aplikasi, dan *error* kompresi *file*. Karena pada *point* kode, nilai pernyataan A4 adalah 58%, B2 adalah 41%, C2 adalah 66%, C4 adalah 58%, C7 adalah 66%, D1 adalah 61%, dan D3 adalah 66%.

4.9. Uji Coba Pengguna

Tujuan dari uji coba pengguna ini untuk menguji apakah algoritma *Lempel-Ziv-Welch* (LZW) ini cocok untuk mengkompresi format citra SVG, BMP, dan TIFF serta pengujian berpusat pada kinerja aplikasi dalam melakukan kompresi dan dekompresi, yang diukur dengan melihat waktu yang diperlukan untuk mengompresi dan mendekompresi *file*. Selain mengukur kinerja aplikasi dari waktu pemrosesan *file*, dilihat juga perbedaan dari ukuran *file* sebelum dan sesudah kompresi serta besar kecilnya persentase rasio kompresi.

Tabel 5. Hasil Uji Coba Kompresi

No	Nama File	Ukuran Sebelum	Ukuran Sesudah	Rasio	Waktu Kompresi
1	bmp 1.bmp	977 KB	864 KB	11,59% lebih kecil	1.16835 ms
2	bmp 2.bmp	2,51 MB	1,57 MB	37,34% lebih kecil	3.14043 ms
3	bmp 3.bmp	3,99 MB	1,16 MB	70,68% lebih kecil	5.80197 ms
4	bmp 4.bmp	11,7 MB	492 KB	95,89% lebih kecil	22.30373 ms
5	bmp 5.bmp	11,8 MB	1,27 MB	89,25% lebih kecil	20.99975 ms
6	svg 1.svg	84,9 KB	166 KB	95,8% lebih besar	0.02115 ms
7	svg 2.svg	63,0 KB	112 KB	78,93% lebih besar	0.02321 ms
8	svg 3.svg	58,6 KB	104 KB	77,65% lebih besar	0.03082 ms
9	svg 4.svg	1,39 MB	3,49 MB	151,3% lebih besar	0.93550 ms
10	svg 5.svg	2,35 MB	5,96 MB	153,12% lebih besar	2.80955 ms
11	tif 1.tif	14,5 MB	644 KB	95,68% lebih kecil	19.42498 ms
12	tif 2.tif	14,6 MB	2,58 MB	82,34% lebih kecil	19.66270 ms
13	tif 3.tif	10,6 MB	1,68 MB	84,26% lebih kecil	14.88861 ms
14	tif 4.tif	12,1 MB	11,0 MB	9,72% lebih kecil	13.91236 ms
15	tif 5.tif	10,5 MB	1,29 MB	87,65% lebih kecil	12.37608 ms

Dari Tabel 5 diatas pada hasil kompresi *file* dengan format BMP dapat dilihat bahwa ukuran *file* setelah kompresi menjadi lebih kecil dari *file* aslinya dan memiliki rasio kompresi yang lebih kecil. Pada hasil kompresi *file* dengan format SVG dapat dilihat bahwa ukuran *file* setelah kompresi menjadi lebih

besar dari *file* aslinya, serta memiliki rasio lebih besar. Pada hasil kompresi *file* dengan format TIFF dapat dilihat bahwa ukuran *file* setelah kompresi menjadi lebih kecil dari *file* aslinya dan memiliki rasio kompresi yang lebih kecil.

Berdasarkan dari 15 uji coba kompresi *file* yang terlampir pada Tabel 5 diatas, dapat disimpulkan bahwa dari hasil uji coba kompresi *file* format BMP dan TIFF dengan algoritma *Lempel-Ziv-Welch* (LZW) lebih efektif dalam hal rasio kompresi dan dapat menghasilkan ukuran *file* yang lebih kecil. Sedangkan untuk kompresi *file* citra SVG dengan algoritma *Lempel-Ziv-Welch* (LZW) tidak efektif dalam pengompresian, yang mana *file* hasil kompresi memiliki ukuran yang lebih besar dari *file* sebelum kompresi. Hal ini dikarenakan citra dengan format SVG berbeda dari citra berformat BMP dan TIFF, karena citra dengan format SVG berbasis *vector*, bukan titik *pixel*. Sehingga algoritma *Lempel-Ziv-Welch* (LZW) tidak efektif digunakan pada citra SVG. Namun dalam hal waktu, kompresi dengan format SVG lebih unggul dari format BMP dan TIFF.

Selanjutnya dilakukan uji coba dekompresi. Hasil uji coba dari dekompresi ditunjukkan pada Tabel 6.

Tabel 6. Hasil Uji Coba Dekompresi

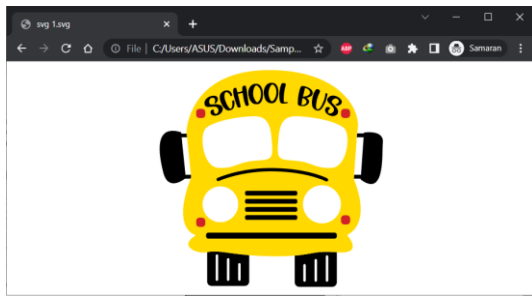
No	Nama File	Ukuran Sebelum	Ukuran Sesudah	Waktu Dekompresi
1	bmp 1.bmp.lzw	864 KB	977 KB	0.05537 ms
2	bmp 2.bmp.lzw	1,57 MB	2,51 MB	0.12172 ms
3	bmp 3.bmp.lzw	1,16 MB	3,99 MB	0.07540 ms
4	bmp 4.bmp.lzw	492 KB	11,7 MB	0.06723 ms
5	bmp 5.bmp.lzw	1,27 MB	11,8 MB	0.10468 ms
6	svg 1.svg.lzw	166 KB	84,9 KB	0.00588 ms
7	svg 2.svg.lzw	112 KB	63,0 KB	0.00526 ms
8	svg 3.svg.lzw	104 KB	58,6 MB	0.00931 ms
9	svg 4.svg.lzw	3,49 MB	1,39 MB	0.27697 ms
10	svg 5.svg.lzw	5,96 MB	2,35 MB	0.46925 ms
11	tif 1.tif.lzw	644 KB	14,5 MB	0.08594 ms
12	tif 2.tif.lzw	2,58 MB	14,6 MB	0.21751 ms
13	tif 3.tif.lzw	1,68 MB	10,6 MB	0.16000 ms
14	tif 4.tif.lzw	11,0 MB	12,1 MB	0.82743 ms
15	tif 5.tif.lzw	1,29 MB	10,5 MB	0.11359 ms

Seperti yang terlihat pada Tabel 6, dapat dilihat dari hasil pengujian dekompresi, *file* kembali ke ukuran semula dan dapat dikatakan proses dekompresi berhasil.

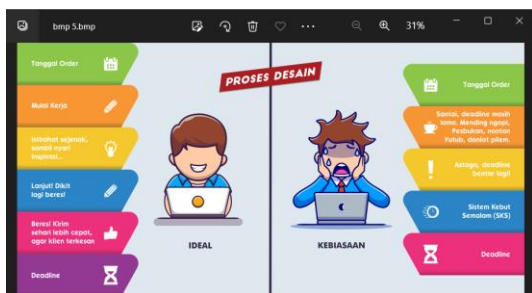
4.10. Tampilan Hasil Kompresi

Hasil tampilan uji coba kompresi *Lempel Ziv Welch* untuk tampilan *file* jenis citra yaitu *file* SVG, BMP, dan TIFF sebelum melakukan proses kompresi masih menampilkan citra yang dapat dilihat, berbeda dengan *file* SVG, BMP, dan TIFF yang sudah

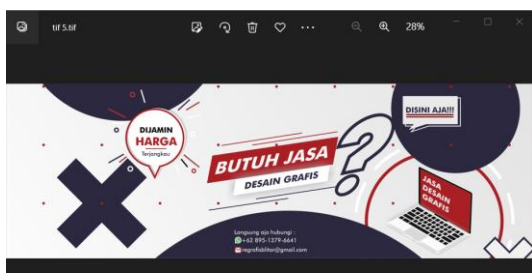
melalui proses kompresi karena telah melalui pemampatan file.



Gambar 12. Tampilan file dari citra SVG sebelum proses kompresi



Gambar 13. Tampilan file dari citra BMP sebelum proses kompresi



Gambar 14. Tampilan file dari citra TIFF sebelum proses kompresi



Gambar 15. Tampilan file dari citra (SVG, BMP, dan TIFF) setelah proses kompresi

Setelah melalui proses kompresi maka tampilan isi dari file citra SVG, BMP, dan TIFF berubah serta tidak dapat terbaca karena semua karakter yang ada telah diubah. Hasil dari proses kompresi yang ada pada Gambar 15 menunjukkan bahwa citra yang telah dikompresi berubah menjadi file teks dengan diberi tambahan ekstensi .LZW.

Aplikasi kompresi citra ini belum bisa menampilkan citra terkompresi secara langsung, hal

tersebut dikarenakan algoritma Lempel Ziv Welch yang bersifat lossless, artinya file terkompresi lossless tidak dapat dibaca atau ditampilkan tanpa melalui proses dekompresi. Sehingga dengan melalui proses dekompresi maka file citra akan kembali seperti semula.

Evaluasi Hasil Rasio dan Waktu Kompresi Lempel Ziv Welch

Rata-rata rasio kompresi Lempel Ziv Welch dan rata-rata waktu pemrosesan kompresi LZW dari berbagai file BMP, file SVG, dan file TIFF ditunjukkan pada Tabel 4.13. Untuk hasil terbaik, jenis file TIFF mencapai rasio rata-rata 71,93%. Rata-rata waktu terbaik dicapai untuk jenis file BMP dengan waktu 0,01 detik, hal ini dikarenakan ukuran file BMP yang relatif kecil membuat proses kompresi lebih cepat.

Tabel 7. Hasil rata-rata rasio dan waktu proses kompresi algoritma Lempel Ziv Welch (LZW)

No	Tipe File	Rasio	Waktu	Rata-Rata Rasio Berdasarkan Tipe File	Rata-Rata Waktu Berdasarkan Tipe File
1.	BMP	11,59% lebih kecil	1.16835 ms	60,95%	0,01 detik
2.	BMP	37,34% lebih kecil	3.14043 ms		
3.	BMP	70,68% lebih kecil	5.80197 ms		
4.	BMP	95,89% lebih kecil	22.30373 ms		
5.	BMP	89,25% lebih kecil	20.99975 ms		
6.	SVG	95,8% lebih besar	0.02115 ms	111,36%	0,06 detik
7.	SVG	78,93% lebih besar	0.02321 ms		
8.	SVG	77,65% lebih besar	0.03082 ms		
9.	SVG	151,3% lebih besar	0.93550 ms		
10.	SVG	153,12% lebih besar	2.80955 ms		
11.	TIFF	95,68% lebih kecil	19.42498 ms	71,93%	0,02 detik
12.	TIFF	82,34% lebih kecil	19.66270 ms		
13.	TIFF	84,26% lebih kecil	14.88861 ms		
14.	TIFF	9,72% lebih kecil	13.91236 ms		
15.	TIFF	87,65% lebih kecil	12.37608 ms		

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan, diantaranya sebagai berikut: Aplikasi yang dibuat untuk mengompresi file citra SVG, BMP, dan TIFF menggunakan algoritma Lempel Ziv Welch telah berhasil diimplementasikan dan berfungsi dengan baik. Tetapi aplikasi ini belum bisa menampilkan citra terkompresi secara langsung, hal tersebut dikarenakan algoritma Lempel Ziv Welch yang bersifat lossless, artinya file terkompresi lossless tidak

dapat dibaca atau ditampilkan tanpa melalui proses dekompresi. Berdasarkan evaluasi hasil rasio dan waktu kompresi dari algoritma *Lempel Ziv Welch*, hasil yang didapat dari rasio rata-rata citra BMP adalah 60,95% dengan nilai besaran antara 11,59% untuk nilai terkecil dan 95,89% untuk nilai tertinggi. Kemudian untuk rasio rata-rata citra SVG adalah 111,36% dengan nilai besaran antara 77,65% untuk nilai terkecil dan 153,12% untuk nilai tertinggi. Untuk rasio rata-rata citra TIFF adalah 71,93% dengan nilai besaran antara 9,72% untuk nilai terkecil dan 95,68% untuk nilai tertinggi. Pada saat yang sama, kecepatan proses kompresi dan dekompresi sebanding dengan ukuran *file*, yang berarti semakin besar *file* yang diproses, semakin lama prosesnya.

Dengan kesimpulan di atas, beberapa saran dibuat sebagai pertimbangan lebih lanjut untuk meningkatkan efisiensi kerja aplikasi kompresi tersebut, berikut ini merupakan saran yang dapat diberikan: Penelitian ini dapat dikembangkan lagi dengan membandingkan algoritma *Lempel-Ziv-Welch* (LZW) dengan algoritma lain seperti *Shannon-fano*, *Run Length Encoding* (RLE), dan *Half-Byte*. Diharapkan penelitian selanjutnya dapat langsung menunjukkan hasil kompresi dalam bentuk citra tanpa harus melalui proses dekompresi lagi.

DAFTAR PUSTAKA

- [1] Mahesa, K., 2018. Rancang Bangun Aplikasi Kompresi Dan Dekompresi Pada Citra Digital Menggunakan Metode Huffman. *Jurnal Processor*, 12(1), pp.948-963.
- [2] Purwanto, H., 2018. Penerapan algoritma huffman pada kompresi file wave. *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, 2(2).
- [3] Rustamaji, H.C., Mariani, M. & Yuwono, B., 2015. Aplikasi Kompresi Data Menggunakan Metode Huffman Statik Pada Perangkat Mobile Berbasis Android. *Telematika: Jurnal Informatika dan Teknologi Informasi*, 11(1).
- [4] Sembiring, A.S., 2018. Analisis Penerapan Metode Lossy Pada Kompresi Citra Steganografi. *MEANS (Media Informasi Analisa dan Sistem)*, 3(2599-3089), p.1.
- [5] Ardiyanto, D. & Purwoto, B.H., 2014. Kompresi citra dengan menggunakan metode delta modulation.
- [6] Laia, Y., 2016. Optimasi Rasio Kompresi Dan Kompleksitas Waktu Kompresi File Teks Menggunakan Algoritma Lempel-ZIV-Welch Dengan Fibonacci Search. *Sinkron: jurnal dan penelitian teknik informatika*, 1(1), pp.6-6.
- [7] Satyapratama, A., Widjianto, W. & Yunus, M., 2015. Analisis Perbandingan Algoritma Lzw Dan Huffman Pada Kompresi File Gambar Bmp Dan Png. *Jurnal Teknologi Informasi: Teori, Konsep, dan Implementasi*, 6(2), pp.69-81.
- [8] Bakara, J., 2020. Implementasi Algoritma LZW dan Kuantisasi Dalam Kompresi Citra Digital. *Pelita Informatika: Informasi dan Informatika*, 5(3).
- [9] Suharso, A., Zaelani, J. & Juardi, D., 2020. KOMPRESI FILE MENGGUNAKAN ALGORITMA LEMPEL ZIV WELCH (LZW). *KOMPUTASI*, 17(2), pp.372-380.
- [10] Hidayat, H., Pamungkas, T. & Zarman, W., 2013. Implementasi Algoritma Kompresi Lzw Pada Database Server. *Komputa: Jurnal Ilmiah Komputer dan Informatika*, 2(1).
- [11] Putra, M.A. & Solichin, A., 2018. Optimasi Web Service dengan Penerapan Algoritma Kompresi LZW: Studi Kasus Aplikasi BluCampus Universitas Budi Luhur. *SKANIKA*, 1(2), pp.455-462.
- [12] Utari, C.T., 2016. Implementasi Algoritma Run Length Encoding Untuk Perancangan aplikasi Kompresi Dan Dekompresi File Citra. *Jurnal Times*, 5(2), pp.24-31.
- [13] Mohammed, S.G., Abdul-Jabbar, S.S. & Mohammed, F.G., 2021, December. Art Image Compression Based on Lossless LZW Hashing Ciphering Algorithm. In *Journal of Physics: Conference Series* (Vol. 2114, No. 1, p. 012080). IOP Publishing.
- [14] Sridhar, A.P. & Lakshmi, P.V., 2021. An Efficient Lossless Medical Data Compression using LZW compression for Optimal Cloud Data Storage. *Annals of the Romanian Society for Cell Biology*, 25(6), pp.17144-17160.